



**Universitat de Lleida  
Escola Politècnica Superior**

**Ingenieria Tècnica en Informàtica de Gestió**

Trabajo Final de Carrera

**Diseño e implementación de un gestor  
para realizar evaluaciones heurísticas**

Autor: Cristina Mur López

Director: Antoni Granollers i Saltiveri

Septiembre 2007

## Agradecimientos

En primer lugar, quiero dar mi más sincero agradecimiento a Toni Granollers, director de este trabajo final de carrera, por haberme guiado y ayudado en la realización del trabajo y haber invertido parte de su tiempo para llevarlo a cabo.

También quiero dar las gracias a Marta Gonzalez por haberme ayudado en la última etapa del proyecto.

Y a Jordi Boncompte por haberme ayudado a resolver los pequeños problemas de implementación que han surgido a lo largo del desarrollo del sistema.

El agradecimiento más especial a mi familia por haberme apoyado siempre y haber hecho posible con su esfuerzo que llegase a donde estoy.

A Diego por haber estado a mi lado y animarme en los momentos difíciles. También a mis amigos y compañeros con los que he pasado los momentos de descanso y han hecho más amena la realización del proyecto.

## Resumen

El trabajo final de carrera consiste en el desarrollo de una aplicación software que permita gestionar evaluaciones heurísticas de diferentes proyectos.

Las evaluaciones heurísticas permiten analizar de manera eficiente la usabilidad de una aplicación.

La aplicación debe ser capaz de almacenar la información correspondiente del proyecto a evaluar, de los evaluadores que realizarán la evaluación, de la heurística empleada en cada evaluación y los resultados de la misma, para más tarde realizar un análisis exhaustivo de la usabilidad de esa aplicación.

# Índice de contenidos

<b>1. Introducción.....</b>	<b>9</b>
1.1. Introducción a la usabilidad.....	9
1.2. Diseño Centrado en el Usuario.....	11
1.3. Fase de Evaluación.....	13
<b>2. Evaluación Heurística.....</b>	<b>15</b>
2.1. Introducción.....	15
2.2. Evaluadores.....	15
2.3. Procedimiento.....	17
2.4. Criterios heurísticos.....	18
<b>3. Análisis de los requisitos y especificación del sistema.....</b>	<b>20</b>
3.1. Análisis etnográfico.....	20
3.2. Análisis de implicados.....	20
3.3. Perfil de usuario.....	21
3.4. Análisis contextual de tareas.....	21
3.5. Objetos.....	21
3.6. Plataforma.....	22
3.7. Perfil del entorno.....	22
3.8. Objetivos del sistema.....	23
3.9. Objetivos de usabilidad.....	23
3.10. Objetivos de accesibilidad.....	24
3.11. Análisis de la competencia.....	25
3.12. Prototipos.....	25
3.12.1. Bocetos.....	25
3.12.2. Prototipos de papel.....	26
3.12.3. Maqueta digital.....	27
3.12.4. Escenarios.....	29
<b>4. Análisis de riesgos, planificación y costes.....</b>	<b>38</b>
4.1. Gestión de riesgos.....	38
4.1.1. Definición y clasificación de riesgos.....	38
4.1.2. Estimación del riesgo.....	39
4.1.3. Control del riesgo.....	40
4.2. Planificación del proyecto.....	41
4.3. Presupuesto.....	43
<b>5. Diseño.....</b>	<b>45</b>
5.1. Análisis jerárquico de tareas.....	45
5.2. Interfaz gráfica.....	52
<b>6. Implementación.....</b>	<b>55</b>
6.1. Plataforma .NET.....	55
6.1.1. ¿Qué es .NET?.....	55
6.1.2. .NET Framework.....	55

6.1.3. Visual Basic .NET.....	57
6.2. XML y DOM.....	57
6.2.1. XML y DTD.....	57
6.2.2. DOM.....	59
6.3. Diseño de la implementación.....	62
6.3.1. Documentos XML.....	62
6.3.2. Clases.....	63
<b>7. Conclusión y futuros trabajos.....</b>	<b>67</b>
<b>8. Bibliografía.....</b>	<b>68</b>
<b>Anexo 0. Plantilla de criterios heurísticos.....</b>	<b>70</b>
0.1. Plantilla de la aplicación.....	70
0.2. Documento Excel utilizado.....	73
<b>Anexo 1. Evaluación de prototipos.....</b>	<b>82</b>
<b>Anexo 2. Manual de usuario.....</b>	<b>87</b>

## Índice de tablas

Tabla 1: Flujo de acontecimientos: Registro del proyecto.....	30
Tabla 2: Flujo de acontecimientos: Registro del evaluador.....	31
Tabla 3: Flujo de acontecimientos: Preparación de la heurística.....	33
Tabla 4: Flujo de acontecimientos: Envío de la heurística.....	34
Tabla 5: Flujo de acontecimientos: Evaluación heurística.....	35
Tabla 6: Flujo de acontecimientos: Envío de los resultados.....	37
Tabla 7: Tabla de riesgos del proyecto.....	40
Tabla 8: Planificación inicial del proyecto.....	42
Tabla 9: Planificación final del proyecto.....	42
Tabla 10: Coste Inventariable.....	43
Tabla 11: Coste Fungible.....	43
Tabla 12: Coste Personal.....	44
Tabla 13: Coste Viajes y dietas.....	44
Tabla 14: Presupuesto final.....	44

## Índice de imágenes

Imagen 1: Secuencia de iteración de las actividades.....	11
Imagen 2: Modelo de Proceso de la Ingeniería de la Usabilidad y la Accesibilidad.....	12
Imagen 3: Proporción de problemas de usabilidad encontrados en una interfaz mediante la evaluación heurística realizada por varios evaluadores.....	16
Imagen 4: Boceto de la pantalla inicial.....	25
Imagen 5: Boceto de la pantalla de información del proyecto.....	25
Imagen 6: Boceto del registro de un proyecto.....	25
Imagen 7: Boceto de la lista de evaluadores.....	25
Imagen 8: Boceto de la heurística.....	26
Imagen 9: Prototipo de la pantalla inicial.....	27
Imagen 10: Prototipo de la pantalla de datos del proyecto.....	27
Imagen 11: Prototipo de la pantalla de evaluadores.....	27
Imagen 12: Prototipo de la pantalla de un criterio.....	27
Imagen 13: Pantalla de la información del proyecto.....	28
Imagen 14: Pantalla de la lista de evaluadores.....	28
Imagen 15: Pantalla de la lista de criterios.....	28
Imagen 16: Pantalla de un criterio.....	28
Imagen 17: Pantalla de las instrucciones de la evaluación.....	28
Imagen 18: Caso de uso: Registro del proyecto.....	30
Imagen 19: Diagrama de secuencia: Registro del proyecto.....	30
Imagen 20: Caso de uso: Registro del evaluador.....	31
Imagen 21: Diagrama de secuencia: Registro del evaluador.....	32
Imagen 22: Caso de uso: Preparación de la heurística.....	32
Imagen 23: Diagrama de secuencia: Preparación de la heurística.....	33
Imagen 24: Caso de uso: Envío de la heurística.....	34
Imagen 25: Diagrama de secuencia: Envío de la heurística.....	34
Imagen 26: Caso de uso: Evaluación heurística.....	35
Imagen 27: Diagrama de secuencia: Evaluación heurística.....	36
Imagen 28: Caso de uso: Envío de los resultados.....	36
Imagen 29: Diagrama de secuencia: Envío de los resultados.....	37
Imagen 30: Tarea: Registrar proyecto.....	45
Imagen 31: Tarea: Modificar proyecto.....	46
Imagen 32: Tarea: Eliminar proyecto.....	46
Imagen 33: Tarea: Registrar evaluador.....	47
Imagen 34: Tarea: Modificar evaluador.....	47
Imagen 35: Tarea: Eliminar evaluador.....	48
Imagen 36: Tarea: Nueva heurística.....	48
Imagen 37: Tarea: Insertar subheurística.....	49
Imagen 38: Tarea: Consultar resultados.....	49
Imagen 39: Tarea: Leer instrucciones.....	50
Imagen 40: Tarea: Hacer evaluación.....	50
Imagen 41: Tarea: Enviar heurística/evaluación.....	51
Imagen 42: Tarea: Descargar heurística/evaluación.....	52
Imagen 43: Lista de proyectos.....	53
Imagen 44: Pestaña Información.....	53

Imagen 45: Pestaña Evaluadores.....	53
Imagen 46: Pestaña Heurística.....	53
Imagen 47: Pestaña Análisis.....	53
Imagen 48: Lista de proyectos.....	54
Imagen 49: Pestaña Información.....	54
Imagen 50: Pestaña Instrucciones.....	54
Imagen 51: Pestaña Evaluación.....	54
Imagen 43: Esquema de componentes dentro de la plataforma .NET Framework.....	56
Imagen 44: Ejemplo de la representación de un documento XML propuesta por DOM.....	60



# 1. Introducción

## 1.1. Introducción a la usabilidad

La interfaz gráfica de una aplicación es un medio para facilitar la interacción entre las personas y el ordenador. A través de la interfaz, los usuarios de la aplicación y el ordenador se comunican mediante la transmisión de información, pero en muchas ocasiones esta comunicación está limitada debido a un diseño pobre de la interfaz. Esto supone que el usuario no entienda la información representada o no pueda realizar alguna actividad.

Por este motivo, existe una disciplina conocida como Interacción Persona-Ordenador (IPO) que estudia el intercambio de información entre el usuario y el ordenador. La IPO se basa en un proceso de diseño centrado en el usuario para conseguir aplicaciones fáciles de utilizar y de aprender, estando directamente relacionada con el diseño, la evaluación y la implementación de las mismas [GLIPO].

Por tanto, los objetivos principales de la IPO son la usabilidad y la accesibilidad. El concepto de usabilidad se define como la rapidez y facilidad con que las personas aprenden a utilizar un sistema interactivo, y accesibilidad como la flexibilidad para acomodarse a las necesidades de cada usuario y a sus preferencias y/o limitaciones [MPIUA].

Existen diferentes definiciones para referirse a la usabilidad. La definición más general es la del estándar ISO 9241-11 [ISO98] que dice:

“El grado en el que un producto puede ser usado por determinados usuarios para conseguir objetivos específicos con eficacia, eficiencia y satisfacción en un contexto específico de uso.”

Como el término usabilidad es asociado al diseño de la interfaz del usuario, existe un estándar para la ingeniería de software ISO/IEC FDIS 9126-1 [ISO00] que define usabilidad como:

“Capacidad de un producto software de ser entendido, aprendido, usado y atractivo para el usuario, cuando es usado bajo unas condiciones específicas.”

Otros autores de renombre como Jacob Nielsen [Nie93] definen la usabilidad como:

“la capacidad de aprendizaje, eficiencia en el uso, facilidad de memorizar, tolerante a errores y subjetivamente satisfactorio”

Una vez definida la usabilidad, es importante saber como medir o valorar la usabilidad de una aplicación. Para que la usabilidad se pueda medir de manera equilibrada, existen una serie de principios definidos [Dix93] en los que se basa la usabilidad:

1. **Facilidad de aprendizaje:** es la facilidad con la que los usuarios son capaces de aprender la utilización de una interfaz y captar los posibles cambios realizados en ella.
2. **Consistencia:** se da cuando los mecanismos de una aplicación se utilizan siempre de la misma manera.
3. **Flexibilidad:** es la diversidad de maneras en las que el usuario y el sistema intercambian información.
4. **Robustez:** se produce cuando la aplicación permite al usuario conseguir sus objetivos sin problemas.
5. **Recuperabilidad:** es la facilidad con que el sistema permite al usuario corregir sus acciones.
6. **Tiempo de respuesta:** se refiere al tiempo que necesita el sistema para mostrar los cambios de estado al usuario.
7. **Adecuación de las tareas:** se da cuando el sistema soporta todas las tareas en el orden deseado por el usuario.
8. **Disminución de la carga cognitiva:** se refiere a que el usuario no tenga que aprender una gran cantidad de detalles.

Se ha podido observar en las definiciones anteriores, que la usabilidad es un pilar fundamental en la IPO que contiene numerosos beneficios. Algunos de los beneficios principales son:

- ◆ Menor tiempo de aprendizaje y mayor asimilación del mismo.
- ◆ Disminución de costes en sistemas de ayuda interna.
- ◆ Menor tiempo para la captura y mantenimiento de datos.
- ◆ Optimización de los costes de diseño, rediseño y mantenimiento.
- ◆ Menor número de errores y del coste invertido en su subsanación.
- ◆ Mejora la imagen y el prestigio.
- ◆ Aumenta las ventas y mejora la calidad del producto.
- ◆ Mayor satisfacción en el uso de las herramientas informáticas por parte de los usuarios.

Como se puede ver, la usabilidad tiene demasiados beneficios como para olvidarse de ella. Por eso, hoy en día el desarrollo del software está unido fuertemente a la usabilidad, que es considerada como un atributo de calidad en el desarrollo del software.

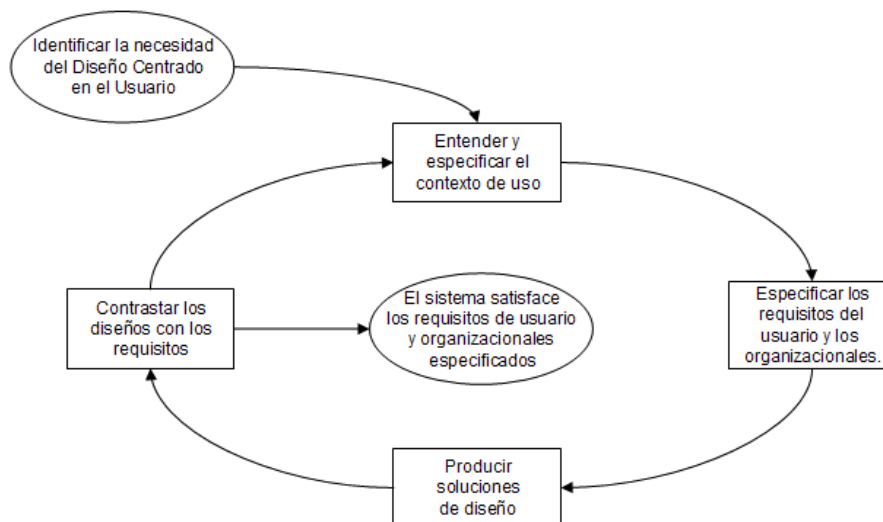
## 1.2.Diseño Centrado en el Usuario

Para que un sistema sea usable y accesible es necesario tener en cuenta a los usuarios de la aplicación en todas las etapas de desarrollo. Como se ha dicho anteriormente, la IPO se basa en un modelo de proceso de diseño centrado en el usuario que se encarga de que el usuario de la aplicación esté implicado directamente en todos los puntos de desarrollo del sistema. El estándar ISO 13407 [ISO99] proporciona una orientación sobre las actividades de diseño centradas en el usuario durante todo el proceso de desarrollo, lo cual describe como una actividad multidisciplinar que incorpora factores humanos, y técnicas y conocimientos ergonómicos, con el fin de conseguir efectividad y eficiencia, y mejorar las condiciones de trabajo para las personas [EG01].

El estándar describe las siguientes actividades [GMPIUA]:

- ♦ Entender y especificar el contexto de uso.
- ♦ Especificar los requisitos de los usuarios y organizativos.
- ♦ Diálogo simple y natural.
- ♦ Producción de soluciones de diseño.

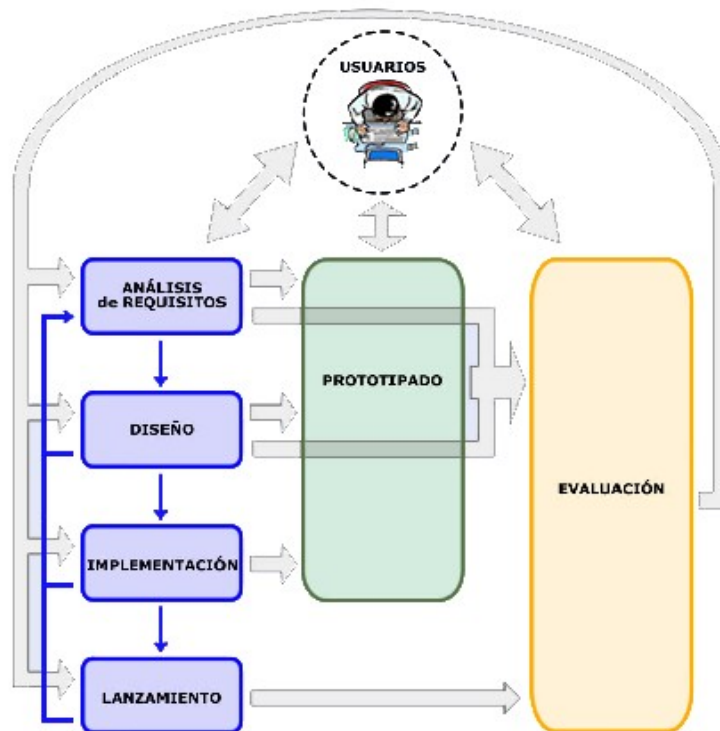
La realización de estas actividades tiene un ciclo iterativo que varía según el tipo de sistema:



*Imagen 1: Secuencia de iteración de las actividades*

Al igual que varía el ciclo iterativo de las actividades, también se ha de tener en cuenta que cada sistema será utilizado por usuarios muy diferentes y que las personas encargadas del desarrollo de la aplicación no tienen porque estar familiarizadas con el entorno donde se va a utilizar. Por eso, es conveniente que el usuario este presente en todas las fases de desarrollo aplicando unas líneas de

trabajo. Estas líneas de trabajo están especificadas y bien organizadas en un modelo de proceso conocido como Modelo de Proceso de la Ingeniería de la Usabilidad y de la Accesibilidad (MPIUiA) representado en el siguiente esquema:



*Imagen 2: Modelo de Proceso de la Ingeniería de la Usabilidad y la Accesibilidad*

Este modelo de proceso basado en el diseño centrado en el usuario tiene siete características fundamentales:

1. **Organización conceptual:** el esquema está organizado en una serie de módulos que determinan la fase de desarrollo en la que se encuentra el sistema.
2. **Tres pilares básicos:** el esquema diferencia claramente los tres pilares básicos en diferentes colores:
  - ♦ Ingeniería del Software: aplicación de un enfoque sistemático, disciplinado y cuantificable hacia el desarrollo, operación y mantenimiento del software (columna de la izquierda de color azul).
  - ♦ Prototipado: implementación parcial pero concreta del diseño de un sistema (columna central de color verde).
  - ♦ Evaluación: comprobación de la usabilidad y accesibilidad de un sistema interactivo (columna de la derecha de color amarillo).

3. **El usuario:** el usuario debe estar presente en todas las fases de desarrollo del sistema.
4. **Iteratividad:** en el esquema se muestran dos tipos de flechas, las delgadas representan el proceso iterativo dentro de la Ingeniería del Software, y las gruesas que representan como el usuario está presente en todas las fases de desarrollo.
5. **Sencillez:** es un proceso sencillo sin dificultad de comprensión.
6. **Adaptado al modelo mental de los equipos multidisciplinares:** es necesario realizar un proceso de desarrollo que entiendan todas las personas de las diferentes áreas de desarrollo.
7. **Flexibilidad:** el equipo de desarrollo, los requisitos del sistema, las particularidades de los usuarios y los resultados de las evaluaciones marcarán cuantas iteraciones deben realizarse, cómo deben hacerse y el flujo de las acciones ha realizar en cada iteración.

### 1.3.Fase de Evaluación

Dentro de las características del modelo de proceso, este trabajo se centrará en la etapa de Evaluación. La evaluación corresponde a la fase de desarrollo en la que se comprueba el grado de usabilidad y accesibilidad del sistema que se está desarrollando. Para realizar esta comprobación existen diferentes técnicas y metodologías que provocarán la constante iteratividad de las fases de proceso. Por este motivo, la evaluación no se realizará solo en la última fase del proceso, sino que estará presente durante todo el ciclo de vida del proceso de desarrollo, utilizándose diferentes métodos de evaluación según la etapa en la que se encuentre el sistema. La evaluación pretende que los diseños de las interfaces se adapten a los requisitos y necesidades de los usuarios finales de la aplicación, y según Dix sus objetivos principales [Dix93] son:

- ♦ Comprobar la extensión de la funcionalidad del sistema: el sistema debe contener todas las especificaciones de los requisitos.
- ♦ Comprobar el efecto de la interfaz en el usuario: el usuario debe realizar las tareas de manera fácil e intuitiva.
- ♦ Identificar cualquier problema específico con el sistema: detectar resultados inesperados y confusión entre los usuarios.

Existen diferentes tipos de evaluaciones según la fase del proceso en la que se encuentre el sistema, ya que unos métodos de evaluación serán más efectivos en unos casos que en otros. Los tipos de evaluación de la usabilidad se pueden clasificar de maneras diferentes:

- ♦ **Lugar de realización**
  - Laboratorio: laboratorio de usabilidad donde el evaluador o los usuarios realizan las comprobaciones del diseño de la interfaz.

- Entorno natural: es el lugar habitual donde se producirá la interacción con el usuario.

- ◆ **Tipo de técnica**

- Inspección: los expertos examinan aspectos de la interfaz del sistema relacionados con la usabilidad y la accesibilidad. Este método permite identificar un gran número de errores de usabilidad a un precio relativamente bajo ya que no implica a los usuarios.
- Indagación: consigue información sobre las necesidades y los gustos de los usuarios y la especificación de los requisitos del sistema. Este método se realiza hablando con el usuario, observándolo y usando el sistema.
- Test: un grupo de usuarios representativos realizan tareas con el sistema y los evaluadores analizan los resultados de la interacción entre el sistema y el usuario.

- ◆ **Automatización**

- Automáticos: son métodos que disponen de software o hardware para comprobar los aspectos a validar. Estos métodos no pueden utilizarse en las fases de inicio de desarrollo.
- Manuales: son métodos que no disponen de software ni de hardware para realizar la evaluación y pueden utilizarse en cualquier fase del desarrollo.

- ◆ **Participantes**

- CON usuarios y/o implicados: los usuarios o implicados intervienen directamente en la evaluación del sistema.
- SIN usuarios y/o implicados: son métodos en los que participan exclusivamente expertos evaluadores con la ayuda de guiones y pautas.

## 2. Evaluación Heurística

### 2.1. Introducción

La evaluación heurística es un método de evaluación de la usabilidad por inspección que fue desarrollado por Nielsen y Molich [NM90]. Su principal objetivo es comprobar la usabilidad existente en el diseño de una aplicación. Este método de evaluación consiste básicamente en un conjunto de expertos evaluadores que inspeccionan la interfaz del sistema siguiendo unas pautas especificadas para detectar los posibles errores de usabilidad del diseño.

La evaluación heurística es un método que se puede utilizar en todo el ciclo de desarrollo, pero es favorable usarla en la etapa de prototipado, ya que detecta una buena cantidad de errores de usabilidad. Debido a sus numerosas ventajas, es uno de los métodos más utilizados en la fase de evaluación:

- ♦ Bajo coste.
- ♦ Detecta una gran cantidad de errores de usabilidad.
- ♦ Número reducido de evaluadores.
- ♦ No es necesaria la interpretación externa de las ideas, porque toda la información está recogida en los informes de los propios evaluadores.
- ♦ Se puede dialogar con los evaluadores sobre los problemas o cuestiones de interés.
- ♦ Puede realizarse en cualquier etapa del proceso de desarrollo.

A pesar de todas las ventajas mencionadas la evaluación heurística tiene un inconveniente a tener cuenta, que es la no participación de usuarios representativos. Estos usuarios podrían detectar fallos o aportar ideas que los expertos evaluadores no pudiesen encontrar.

### 2.2. Evaluadores

Los evaluadores son las personas encargadas de llevar a cabo la evaluación heurística. Su trabajo consiste en inspeccionar el diseño de la interfaz para poder detectar los errores de usabilidad y aportar ideas para resolver estos errores.

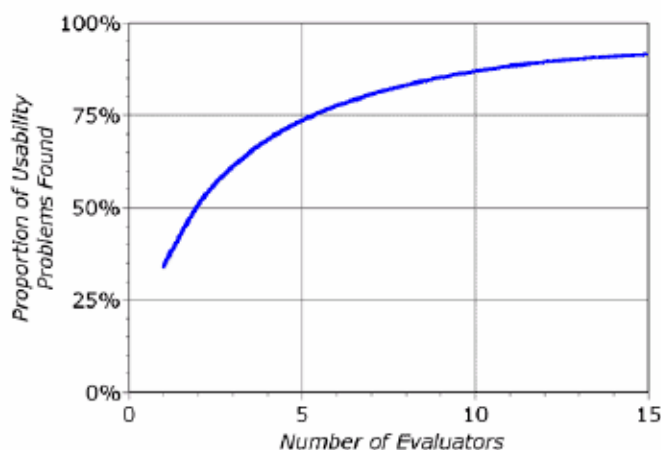
La elección de los evaluadores que participen en la evaluación heurística no es una tarea sencilla. El perfil de los evaluadores es un aspecto importante a la hora de seleccionarlos. Los evaluadores tienen que tener experiencia en la evaluación de la usabilidad y en el diseño de interfaces, además de conocer todo lo referente sobre el sistema en desarrollo. A pesar de que los evaluadores deben tener experiencia en la usabilidad, algunos autores proponen implicar a los

usuarios del sistema y a los desarrolladores de aplicaciones como posibles evaluadores, basándose en el Diseño Centrado en el Usuario [Nie92, MMPG98, GPL04].

A continuación se explican las características de los posibles evaluadores:

- ♦ **Expertos en usabilidad:** son los más adecuados para este método de evaluación ya que pueden detectar los problemas de usabilidad según los criterios heurísticos. Por el contrario, no tienen experiencia en el dominio de uso de la aplicación por lo que hay errores que no detectarán.
- ♦ **Desarrolladores de aplicaciones:** no son muy recomendados ya que pueden no tener conocimientos de la usabilidad y tienden a centrarse en los problemas técnicos del sistema.
- ♦ **Usuarios de la aplicación:** los usuarios sin experiencia son bastante confusos, ya que no saben expresar claramente los problemas detectados ni las posibles mejoras a esos problemas. Sin embargo, si el usuario conoce o ha utilizado previamente la aplicación, o ha sido partícipe en su desarrollo, puede detectar problemas de usabilidad de manera eficaz.

Elegir a una sola persona como evaluador es poco conveniente, puesto que según el tipo de evaluador que se escoja éste detectará unos problemas u otros. Según un estudio realizado por Nielsen y Landauer [NL93], se podría encontrar la mayoría de los problemas de usabilidad utilizando entre tres y cinco evaluadores. Como se puede observar en el gráfico siguiente, la inclusión de más expertos no garantiza una mejora en los resultados.



*Imagen 3: Proporción de problemas de usabilidad encontrados en una interfaz mediante la evaluación heurística realizada por varios evaluadores.*



## 2.3.Procedimiento

El primer paso para realizar una evaluación heurística es elegir a los evaluadores más adecuados (en el apartado anterior se han explicado los diferentes tipo de evaluadores que se pueden seleccionar). A cada evaluador se le debe explicar el funcionamiento del sistema y se le informa sobre el dominio de la aplicación. Es conveniente que el evaluador interactúe con el sistema al menos dos veces antes de empezar la evaluación para familiarizarse con el entorno.

Las sesiones de evaluación suelen durar entre una y dos horas aproximadamente para cada parte de la interfaz.

La realización de la evaluación consiste en que el evaluador inspeccione detalladamente todos los aspectos de la interfaz especificados por los criterios heurísticos. Es muy importante que durante la evaluación, los evaluadores no se comuniquen entre ellos para no influir en sus decisiones. Si los usuarios del sistema van a disponer de manuales o ayudas en línea, los evaluadores también tienen que disponer de ellos para evaluar el conjunto total del producto.

La información obtenida por los evaluadores puede recogerse de maneras diferentes:

- ♦ **Informe:** Durante la evaluación el evaluador anota todos los problemas detectados y las observaciones. Una vez terminada la inspección, el evaluador redacta un informe basado en las anotaciones con los problemas detectados, las observaciones necesarias y las posibles mejoras del diseño. Finalmente, el responsable de la evaluación recoge los informes de todos los evaluadores para analizar los resultados globalmente. La ventaja del informe es que permite presentar un escrito formal sobre la evaluación, pero requiere un esfuerzo añadido para los evaluadores y para el responsable de la evaluación la necesidad de leerlo y crear un documento que contenga el trabajo de todos los evaluadores.
- ♦ **Expresión oral:** Mientras realiza la evaluación, el evaluador dicta a un observador los problemas que encuentra y sus opiniones. De esta manera expresa una serie de errores que hubiera pasado por alto si él mismo los tuviera que anotar, y supone una disponibilidad casi inmediata de los resultados de la evaluación. Por otra parte, la necesidad de un observador en cada sesión de evaluación supone un gasto adicional.
- ♦ **Categorías:** Antes de empezar la evaluación, los evaluadores se ponen de acuerdo en las categorías en las que tienen que clasificar los problemas detectados. Esta manera hace que el análisis final sea más simple, pero pueden perderse algunos detalles importantes.
- ♦ **Checklist de criterios:** Es una lista de criterios heurísticos que el evaluador tiene que analizar cuando hace la inspección de la interfaz, y puntuar cada una de las preguntas según el grado de usabilidad. También permite puntuar la gravedad de cada problema según la frecuencia con la que se produce, el impacto que provoca cuando sucede y la persistencia del mismo. La puntuación de las preguntas se basa en una escala de severidad determinada previamente por el responsable de la evaluación. Es una manera muy utilizada debido a su comodidad y rapidez de realización.

Una vez que el responsable tiene el resultado global de la evaluación heurística se reúne con los evaluadores para discutir las posibles soluciones a los problemas hallados.

## 2.4. Criterios heurísticos

Los criterios heurísticos son los principios de usabilidad en los que se basan las evaluaciones heurísticas para inspeccionar todos los aspectos que debe cumplir el sistema. Estos criterios son la base de los checklist, que están basados íntegramente en subheurísticas. Las subheurísticas son las preguntas que detallan las características de los criterios, que suelen ser bastante generales.

Existe una serie de criterios recogidos por Nielsen [NM94] que deben de tenerse en cuenta a la hora de realizar una evaluación heurística. A estos criterios se les conoce como *Las 10 reglas heurísticas de usabilidad de Nielsen*:

1. **Visibilidad del estado del sistema.** El sistema debe mantener siempre informados a los usuarios del estado del sistema, con una realimentación apropiada y en un tiempo razonable.
2. **Relación entre el sistema y el mundo real.** El sistema debe hablar el lenguaje del usuario, con palabras, frases y conceptos familiares, en lugar de utilizar términos orientados al sistema. Debe seguir las convenciones del mundo real para que la información aparezca de forma natural y lógica.
3. **Control y libertad para el usuario.** Los usuarios pueden elegir funciones del sistema por error, por lo que deben disponer siempre de una salida de emergencia claramente destacada sin necesidad de diálogo.
4. **Consistencia y estándares.** Los usuarios no deben tener que preguntarse si las diversas palabras, situaciones, o acciones significan la misma cosa. A la hora de implementar el sistema hay que seguir las normas y convenciones para evitar que el usuario se desoriente.
5. **Prevención de errores.** Un diseño cuidadoso que evite que un problema ocurra es mejor que buenos mensajes de error. Eliminar las condiciones de errores propensos o comprobarlas y mostrar a los usuarios una opción de confirmación antes de ejecutar la acción.
6. **Reconocimiento mejor que memoria.** Minimizar la carga de memoria del usuario creando objetos, acciones y opciones visibles. El usuario no debería tener que memorizar información. Las instrucciones para el uso del sistema deben ser visibles o fácilmente recuperable cuando sea apropiado.
7. **Flexibilidad y eficiencia de uso.** Aceleradores -- no visto por el usuario principiante -- pueden acelerar a menudo la interacción para el usuario experto tal que el sistema puede abastecer a los usuarios inexpertos y expertos. Permitir que los usuarios adapten acciones frecuentes.
8. **Diálogos estéticos y diseño minimalista.** Los diálogos no deben contener información que sea irrelevante o raramente necesitada. Cada unidad adicional de información en un diálogo compite con las unidades relevantes de información y disminuye su visibilidad relativa.

9. **Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de los errores.** Los mensajes de error se deben expresar en lenguaje plano (no código), indican exactamente el problema y sugieren constructivamente una solución.
10. **Ayuda y documentación.** Aunque es mejor si el sistema puede utilizarse sin documentación, puede ser necesario disponer de ayuda y documentación. Cualquier información debe ser fácil de buscar, centrada en la tarea del usuario, enumerar los pasos concretos a seguir, y no ser demasiado extensa.

Las 10 reglas de usabilidad de Nielsen son de las más conocidas, pero existen otras guías de criterios de otros autores que también son utilizadas.

Ben Schneiderman, en 1986, presenta sus “Ocho reglas de oro para el diseño de Interfaces” [Sh86], que son entendidas como principios heurísticos generales porque pueden guiar tanto en el diseño de un sistema como en la evaluación de usabilidad.

En el año 1994, Larry Constantine propuso una serie de principios de usabilidad para el desarrollo de interfaces capaces de guiar la definición de una evaluación heurística.

En 1996, Keith Instone escribió un informe “Usability Engineering on the Web”, donde se describían una serie de principios a tener en cuenta en el diseño de una interfaz.

Deborah Mayhew, en 1999, escribió en su libro “The Usability Engineering Lifecycle” [May99] una serie de principios para incorporar la usabilidad al ciclo de vida de un producto.

También Bruce ((Tog)) Tognazzini, en 2003, muestra en su sitio web “Ask Tog” [Tog03], sus principios básicos para el diseño de interfaces.

## **3. Análisis de los requisitos y especificación del sistema**

El análisis de requisitos nos proporciona información sobre los objetivos y funcionalidades del sistema. Determina, enumera y clasifica todas las características, capacidades y restricciones que este debe cumplir o al que se ve sometido. Esta fase garantiza un mayor grado de usabilidad y accesibilidad en el diseño del producto.

Los requisitos son descripciones de como el sistema debe comportarse, la información acerca del dominio de aplicación, las restricciones operativas del sistema o las especificaciones de las propiedades o atributos del sistema. Es importante recoger correctamente los requisitos del sistema, ya que en la mayoría de las veces se produce una mala comunicación entre los futuros usuarios y las personas que desarrollan el sistema, debido a que los usuarios no expresan adecuadamente sus necesidades y pensamientos.

### **3.1. Análisis etnográfico**

El análisis etnográfico proporciona información sobre dónde y cómo los usuarios realizan las tareas relacionadas con el sistema.

Las evaluaciones heurísticas se realizan en laboratorios de usabilidad. Un laboratorio de usabilidad es una sala con ordenadores donde los expertos y los usuarios realizan evaluaciones del sistema.

El responsable de la evaluación se sienta delante del ordenador y examina el diseño a evaluar. Una vez se ha familiarizado con el entorno interactivo, analiza las preguntas de una plantilla de heurísticas predefinida y elige las más adecuadas para la evaluación heurística del proyecto. Cuando ya tiene preparada la heurística se la pasa a los evaluadores. Los evaluadores examinan la interfaz individualmente y después comienzan la evaluación. Mientras realizan la evaluación anotan por escrito la valoración de las preguntas en la hoja de la heurística y comunican sus observaciones o impresiones por escrito o hablando directamente con el responsable. No es necesario que los evaluadores realicen la evaluación al mismo tiempo ni en el mismo lugar. Una vez que el responsable dispone de todos los resultados de las evaluaciones, realiza el análisis global de los resultados que determinará el grado de usabilidad de la interfaz.

### **3.2. Análisis de implicados**

Los implicados son personas u organizaciones que serán afectadas por el sistema y que tienen influencia directa o indirecta en el análisis de los requisitos. Muchas veces los implicados son difíciles de detectar y pueden influir de manera positiva o negativa en el proyecto.

Los implicados en este sistema son el responsable de la evaluación, los evaluadores, los responsables de diseño y desarrollo del sistema, y los responsables de su implantación y mantenimiento.

### 3.3. Perfil de usuario

El sistema será utilizado por dos tipos de usuarios: el responsable de la evaluación y los evaluadores expertos. El responsable de la evaluación es una persona con experiencia en usabilidad y en realizar evaluaciones heurísticas. Los evaluadores expertos son personas especializadas en usabilidad y en el conocimiento del diseño de interfaces.

### 3.4. Análisis contextual de tareas

El análisis contextual de tareas es un estudio de las diferentes tareas que realiza el usuario durante la evaluación heurística. Las tareas principales para este proyecto son:

- ♦ **Escoger a los evaluadores.** El responsable de la evaluación escoge a un número de evaluadores expertos en usabilidad para realizar la evaluación heurística.
- ♦ **Preparar la heurística.** El responsable selecciona las subheurísticas más adecuadas al proyecto a evaluar. Las subheurísticas las puede definir él o sacarlas de una plantilla.
- ♦ **Evaluación.** Los evaluadores realizan la evaluación rellenando la heurística que les ha pasado el responsable. Antes de realizar la evaluación, es recomendable que los evaluadores interactúen con el sistema.
- ♦ **Análisis.** El responsable realiza un estudio de usabilidad con las evaluaciones de los evaluadores.

### 3.5. Objetos

Para realizar evaluaciones heurísticas los implicados utilizan una serie de objetos para llevar a cabo su tarea. Tanto el responsable de la evaluación como los evaluadores utilizan un ordenador, hojas de heurísticas, impresora, bolígrafo y cuaderno de notas.

Este proyecto ha tomado como hojas de heurísticas un archivo Excel donde se almacena toda la información de la evaluación y en el que se ha basado el desarrollo de esta aplicación. También una serie de documentos en los que se explican las reglas de la evaluación y los criterios de la heurística. (Ver Anexo 0)

### 3.6.Plataforma

La plataforma tecnológica requerida por la aplicación define una serie de posibilidades o restricciones según sus características.

#### Hardware

Requisitos mínimos de hardware para el buen funcionamiento del sistema desarrollado:

- ◆ Procesador Intel Pentium a 1.6 Ghz
- ◆ Monitor color 15.4" y resolución 1024x768
- ◆ 1 GB de memoria RAM
- ◆ Disco duro 80 GB
- ◆ Ratón y teclado
- ◆ Impresora

#### Software

Software utilizado para el desarrollo de la aplicación:

- ◆ Microsoft Windows XP Professional
- ◆ Microsoft Visual Studio 2005 Professional
- ◆ Tecnología XML y DOM
- ◆ Microsoft Office PowerPoint XP

### 3.7.Perfil del entorno

Las evaluaciones heurísticas se realizan en laboratorios de usabilidad habilitados con ordenadores donde los implicados en las evaluaciones llevan a cabo sus tareas. Los laboratorios son sitios tranquilos sin ninguna molestia externa. Un laboratorio de usabilidad tiene principalmente dos partes, una sala habilitada con ordenadores que generalmente tiene una pared con una ventana de cristal para que los observadores pueden observar a los usuarios y a los expertos mientras realizan sus tareas sin que se produzca ningún tipo de intervención, y otra sala habilitada con una mesa con sillas donde se organizan las reuniones necesarias con los implicados en el proyecto.

### 3.8. Objetivos del sistema

Una vez explicado qué es y cómo funciona una evaluación heurística, se deben especificar los objetivos que tendrá que cumplir la aplicación para gestionar correctamente las evaluaciones heurísticas de diferentes proyectos. Los principales objetivos especificados son:

- ◆ Guardar la información básica de los proyectos.
- ◆ Guardar la información básica de los evaluadores que realizarán la evaluación.
- ◆ Seleccionar las subheurísticas de la evaluación.
- ◆ Enviar la evaluación heurística a los evaluadores.
- ◆ Introducir la valoración de cada subheurística con posibles comentarios.
- ◆ Recibir los resultados de la evaluación heurística.
- ◆ Analizar los resultados de los evaluadores y realizar un informe.  
Este objetivo no es específico de este proyecto, pero sí fundamental en las posibles extensiones del mismo.

### 3.9. Objetivos de usabilidad

Se debe recordar que la interfaz del sistema ha de ser usable, por lo que es necesario determinar los objetivos de usabilidad específicos para la aplicación. El análisis de estos objetivos se basará en los principios básicos de usabilidad.

- ◆ Facilidad de aprendizaje:
  - Los usuarios serán capaces de utilizar la aplicación sin ningún tipo de aprendizaje.
  - El diseño de la interfaz tiene que ser simple, fácil de aprender y utilizar, con funcionalidades bien definidas.
- ◆ Consistencia:
  - Tanto el módulo del responsable como el módulo de los evaluadores tendrá el mismo diseño de interfaz
  - Se evitarán las palabras subrayadas para no confundir con hipervínculos.
  - Todos los proyectos registrados en la aplicación funcionarán de la misma manera sin cambios específicos en el diseño.

- ◆ Flexibilidad:
  - Se evitará la escritura por parte del usuario excepto en los casos que sea imprescindible.
  - El sistema funcionará mediante botones de selección u otros tipos de navegación que no requieran escritura.
- ◆ Robustez:
  - Se utilizará un sistema de base de datos que no requiera un tipo de software específico.
- ◆ Recuperabilidad:
  - El usuario puede equivocarse en la realización de las tareas. Debe existir una retroalimentación apropiada del sistema.
- ◆ Tiempo de respuesta:
  - El tiempo en que se carga la aplicación no debe exceder de 15 segundos.
- ◆ Adecuación de las tareas:
  - Algunas de las tareas a realizar requieren un orden específico. El sistema soportará el orden deseado por el usuario en las tareas que no requieran otras tareas previas.
- ◆ Disminución de la carga cognitiva:
  - El usuario alcanzará sus objetivos navegando como máximo por 2 pantallas diferentes.
  - El seguimiento de las tareas es intuitivo y puesto que el usuario será experto en evaluaciones heurísticas, no será necesario que recuerde detalles.

### **3.10. Objetivos de accesibilidad**

- ◆ Se utilizará un color de fondo claro para no dificultar la lectura.
- ◆ Los botones estarán bien visibles y describirán sus acciones.
- ◆ El usuario sabrá en todo momento donde se encuentra.



### 3.11. Análisis de la competencia

No se tiene constancia de una aplicación similar. El único sistema parecido que se utiliza son los archivos Excel comentados anteriormente, en los que se basa esta aplicación.

### 3.12. Prototipos

Los prototipos son documentos o sistemas que simulan el sistema final a desarrollar. Estas simulaciones se basan en diseños o tareas que más tarde se tendrán que implementar.

#### 3.12.1. Bocetos

Los bocetos ayudan a representar las primeras ideas tanto de la interfaz del sistema como de sus funcionalidades. Se utilizan en la etapa inicial del diseño, incluso antes de realizar el análisis de requisitos del sistema. Se realizan de una forma rápida, utilizando papel y lápiz.

Las primeras ideas para el diseño de esta aplicación han sido las siguientes:

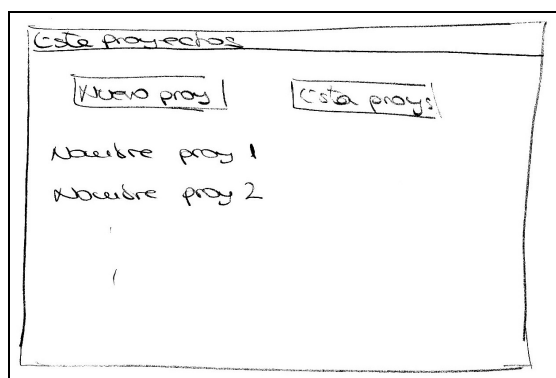


Imagen 4: Boceto de la pantalla inicial

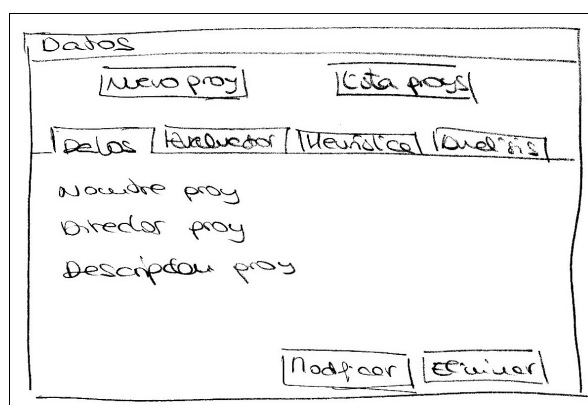


Imagen 5: Boceto de la pantalla de información del proyecto

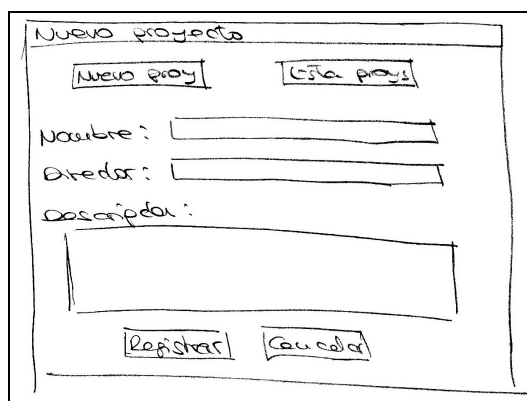


Imagen 6: Boceto del registro de un proyecto

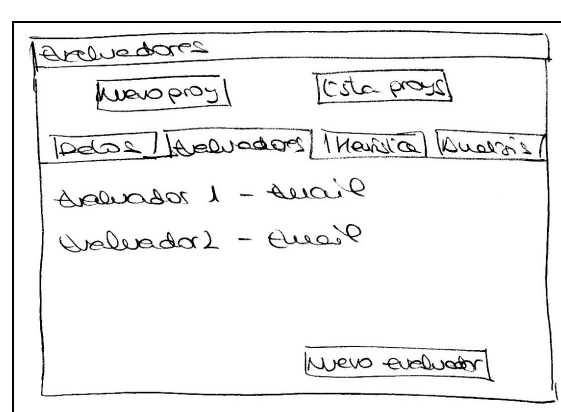


Imagen 7: Boceto de la lista de evaluadores

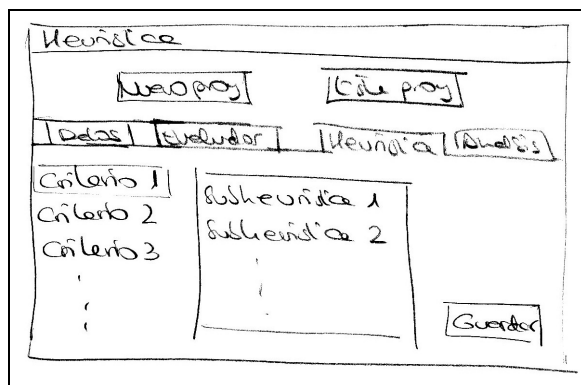


Imagen 8: Boceto de la heurística

### 3.12.2. Prototipos de papel

El objetivo principal de los prototipos de papel es la evaluación del mismo por parte de los usuarios. Con el prototipo de papel se podrá realizar la primera evaluación del proceso de desarrollo del sistema, por lo que es una actividad fundamental para el objetivo de usabilidad.

El método a seguir para la creación del prototipo es muy sencillo. Con lápiz y papel podemos crear un prototipo lo suficientemente útil para realizar la primera evaluación con los usuarios. Habitualmente los prototipos de papel se diseñan con una serie de pestañas añadidas a las hojas de los diseños para poder navegar por la interfaz según les indica la aplicación motorizada. De esta manera, no solo evalúan el diseño, sino también la interacción con el sistema.

El sistema será utilizado por diferentes usuarios que tendrán los mismos criterios de organización de la información, de la estructura del diseño o de gustos en formas y colores. La participación de los usuarios en las evaluaciones de los prototipos forma parte del “proceso de diseño centrado en el usuario” que se ha explicado en la unidad anterior.

Una vez termina la evaluación del prototipo de papel debemos modificar los errores y mejoras que se han determinado durante la evaluación de los usuarios. Así mismo, se ha de modificar también el análisis de requisitos ya que los usuarios son capaces de expresar mejor sus ideas observando un diseño concreto.

Algunas de las pantallas del prototipo de papel son:

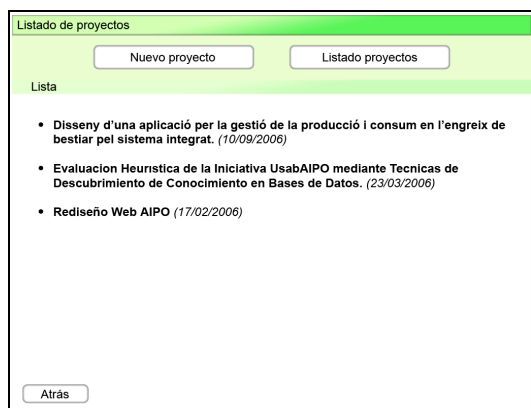


Imagen 9: Prototipo de la pantalla inicial

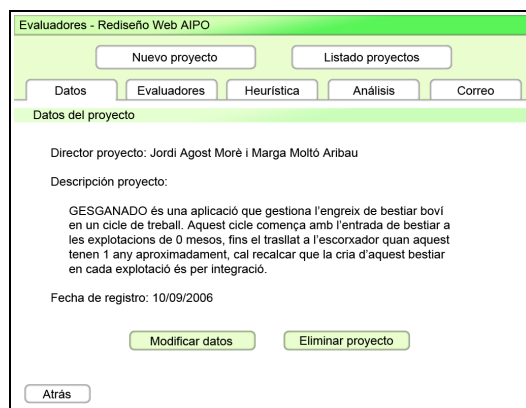


Imagen 10: Prototipo de la pantalla de datos del proyecto



Imagen 11: Prototipo de la pantalla de evaluadores

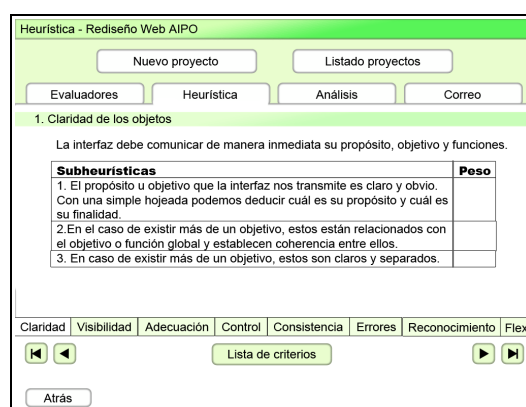


Imagen 12: Prototipo de la pantalla de un criterio

### 3.12.3. Maqueta digital

Las maquetas digitales son simulaciones de los prototipos de papel en formato digital. Este tipo de prototipos es adecuado para hacer las primeras evaluaciones con los usuarios, ya que la interacción con la maqueta será similar a la interacción con el sistema final.

Algunas de las pantallas de la maqueta digital son:

Imagen 13: Pantalla de la información del proyecto

Imagen 14: Pantalla de la lista de evaluadores

Imagen 15: Pantalla de la lista de criterios

Subheurísticas	Peso
1. El propósito u objetivo que la interfaz nos transmite es claro y obvio. Con una simple hoja se podemos deducir cuál es su propósito y cuál es su finalidad.	
2. En el caso de existir más de un objetivo, estos están relacionados con el objetivo o función global y establecen coherencia entre ellos.	
3. En caso de existir más de un objetivo, estos son claros y separados.	

Imagen 16: Pantalla de un criterio

Imagen 17: Pantalla de las instrucciones de la evaluación

### 3.12.4. Escenarios

Los escenarios son una forma apropiada de reflejar las historias entre las personas y sus actividades. Es importante que un escenario contenga la mayoría de las situaciones que se puedan producir durante la interacción del usuario con el sistema.

Estas situaciones se pueden representar de diferentes maneras:

- ◆ Lenguaje natural: se describen las situaciones mediante una narración escrita.
- ◆ Storyboards: describen las situaciones mediante secuencias de dibujos que representan las acciones que el usuario puede realizar mientras interactúa con la interfaz.
- ◆ Videos: grabaciones de video que representan las acciones de los usuarios con el sistema.
- ◆ Casos de uso de UML: las situaciones se describen mediante secuencias de interacciones entre el sistema y los actores utilizando una notación formal.

En este caso, se ha optado por la descripción de los escenarios mediante los casos de uso.

Este método es fácil de entender debido a la notación formal que evita las interpretaciones ambiguas. Los actores de las situaciones son representados por símbolos de forma humana y los casos de uso por elipses.

A continuación especificamos cada caso de uso, junto con el flujo de acontecimientos y el diagrama de secuencias correspondiente.

Los casos de uso del proyecto son:

- ◆ Registro del proyecto
- ◆ Registro del evaluador
- ◆ Preparación de la heurística
- ◆ Envío de la heurística
- ◆ Realización de la evaluación
- ◆ Envío de los resultados
- ◆ Análisis de los resultados

### Caso de uso: Registro del proyecto

Para realizar la evaluación heurística de un proyecto es necesario registrar el proyecto a evaluar. La aplicación podrá tener registrados más de un proyecto. En el registro del proyecto se introducirán el título del proyecto, el/los director/es del proyecto, una breve descripción que explique los objetivos del mismo y su entorno de trabajo, y la fecha de registro. Cada proyecto registrado tendrá su propia evaluación heurística y sus evaluadores.

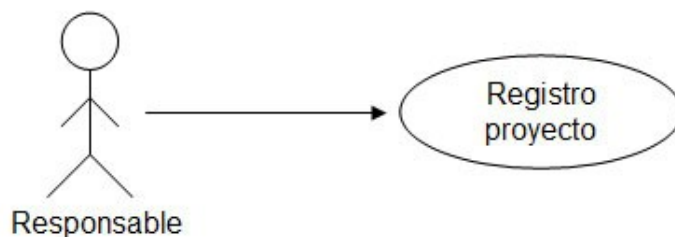


Imagen 18: Caso de uso: Registro del proyecto

Responsable	Sistema
1. El responsable quiere registrar un proyecto nuevo.	
2. El responsable introduce los siguientes datos del proyecto: título, director, descripción y fecha de registro.	
	3. El sistema almacena los datos del proyecto.

Tabla 1: Flujo de acontecimientos: Registro del proyecto

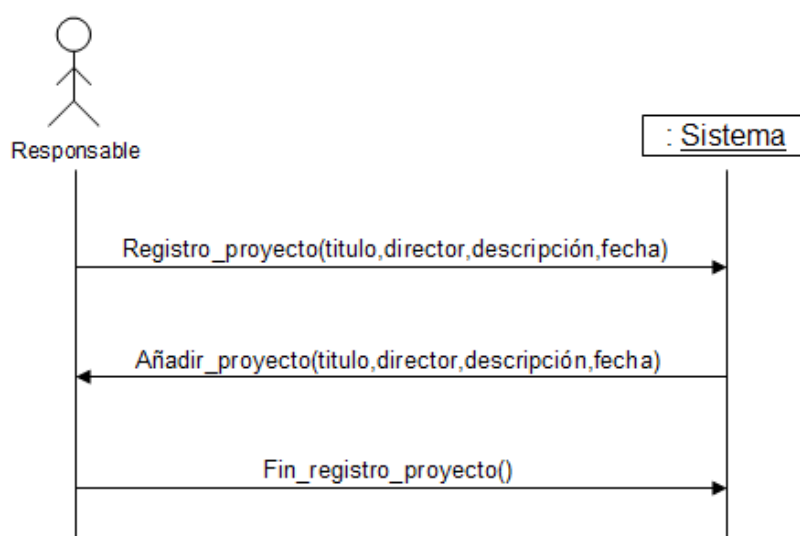


Imagen 19: Diagrama de secuencia: Registro del proyecto

**Pre:** Existe un proyecto para registrar.

**Post:** Se ha registrado un proyecto nuevo con un título, director/es, descripción y fecha de registro.

**Excepción:** Existe un proyecto con este título.

### Caso de uso: Registro del evaluador

Los evaluadores son quienes llevan a cabo la evaluación heurística. Cada proyecto tendrá registrados los evaluadores que realizarán su evaluación. La información necesaria para el registro es el nombre del evaluador, su correo electrónico y su teléfono. El correo electrónico se usará como el identificador de cada evaluador.

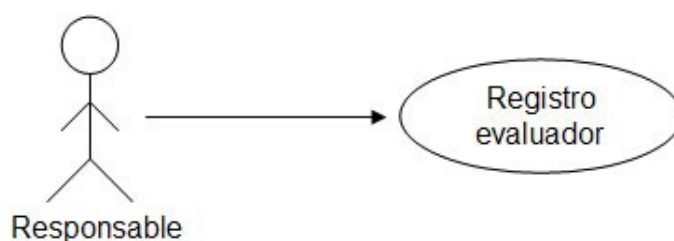


Imagen 20: Caso de uso: Registro del evaluador

Responsable	Sistema
1. El responsable quiere registrar un responsable nuevo.	
2. El responsable introduce los siguientes datos del evaluador: nombre, correo electrónico y teléfono.	
	3. El sistema almacena los datos del evaluador.

Tabla 2: Flujo de acontecimientos: Registro del evaluador

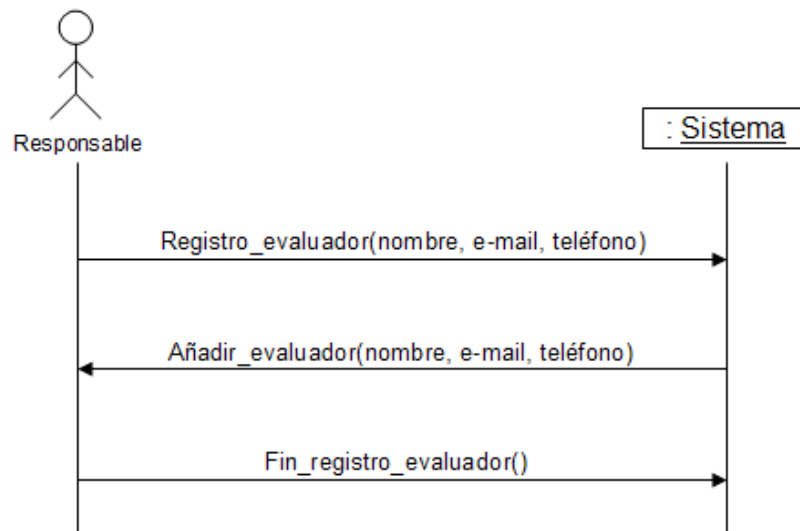


Imagen 21: Diagrama de secuencia: Registro del evaluador

**Pre:** Existe un evaluador para registrar.

**Post:** Se ha registrado un evaluador nuevo con nombre, correo electrónico y teléfono.

**Excepción:** Existe un evaluador con este nombre.

### Caso de uso: Preparación de la heurística

La evaluación heurística consta de una serie de preguntas según los criterios heurísticos para analizar los aspectos del diseño de un sistema. La preparación de la heurística consiste en seleccionar las preguntas más adecuadas de cada criterio para poder evaluar todos los aspectos del proyecto.

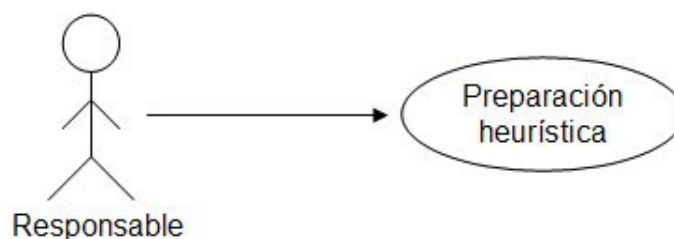


Imagen 22: Caso de uso: Preparación de la heurística



Responsable	Sistema
1. El responsable quiere preparar la heurística de un proyecto determinado.	
	2. El sistema muestra la lista de criterios heurísticos.
3. El responsable inicia el recorrido por cada uno de los criterios.	
	4. El sistema muestra uno por uno los criterios heurísticos con las subheurísticas correspondientes.
5. El responsable selecciona las subheurísticas más adecuadas para la evaluación del proyecto.	
	6. El sistema almacena las preguntas seleccionadas para ese proyecto en concreto.

Tabla 3: Flujo de acontecimientos: Preparación de la heurística

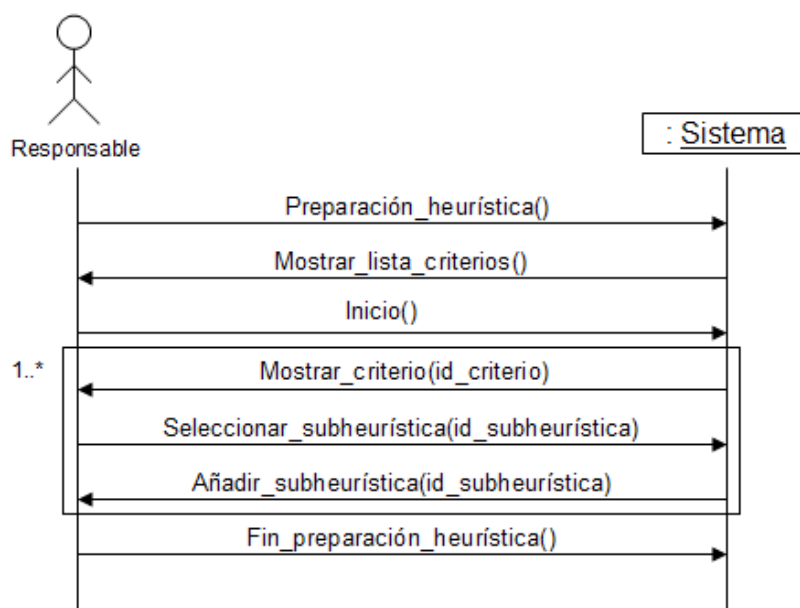


Imagen 23: Diagrama de secuencia: Preparación de la heurística

**Pre:** El proyecto está registrado y sin una heurística realizada.

**Post:** Se han guardado en el sistema las subheurísticas seleccionadas para la evaluación del proyecto.

**Excepción:** No existe ningún proyecto. El proyecto ya tiene una heurística preparada.

### Caso de uso: Envío de la heurística

Una vez preparada la heurística, ésta se envía a todos evaluadores registrados del proyecto para que puedan realizar la evaluación. Los evaluadores recibirán la heurística en su correo electrónico.



Imagen 24: Caso de uso: Envío de la heurística

Responsable	Sistema	Evaluador
1. El responsable quiere enviar la heurística a los evaluadores.		
	2. El sistema envía la heurística al correo de cada evaluador.	
		3. Los evaluadores reciben heurística en su correo electrónico.

Tabla 4: Flujo de acontecimientos: Envío de la heurística

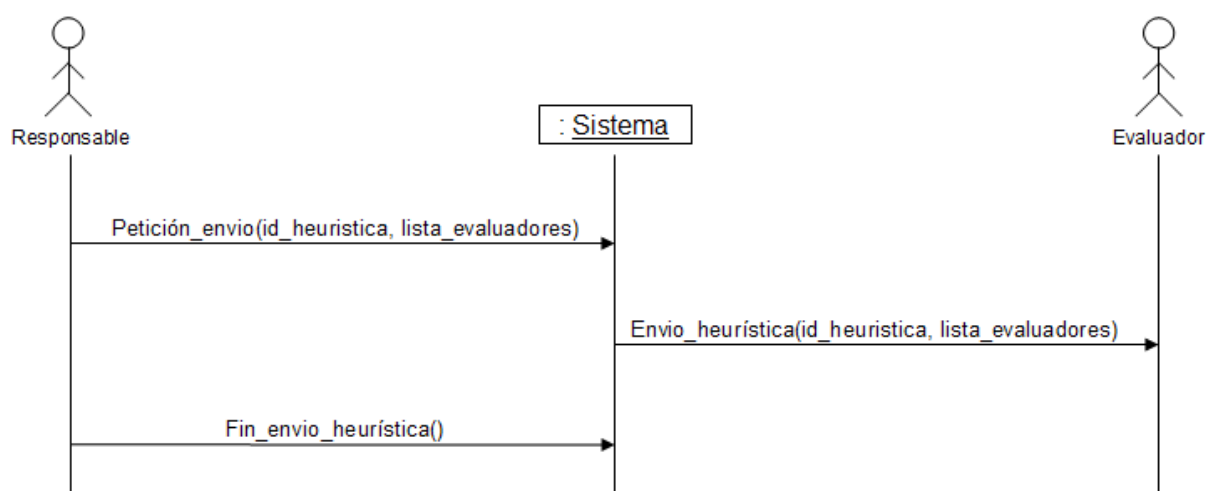


Imagen 25: Diagrama de secuencia: Envío de la heurística

**Pre:** Existe un proyecto con una heurística preparada y existe al menos un evaluador.

**Post:** El evaluador recibe la heurística y ya puede realizar la evaluación.

**Excepción:** No existe ningún proyecto. No existe una heurística para el proyecto. No existe ningún evaluador.

### Caso de uso: Evaluación heurística

Una vez recibida la heurística, los evaluadores ya pueden comenzar la evaluación. Los evaluadores analizan la interfaz del proyecto según las preguntas de la heurística. Cada pregunta recibe una puntuación según el contenido y la estructura del diseño y se pueden hacer observaciones de cada criterio.

El valor de las puntuaciones está definido en:

- 0: No es un problema.
- 1: Problema que no necesita dedicarle tiempo.
- 2: Problema mínimo de usabilidad. La solución es de baja prioridad.
- 3: Problema grave de usabilidad. La solución es de alta prioridad.
- 4: Problema crítico de usabilidad. La solución debe ser inmediata.
- 5: No se aplica.

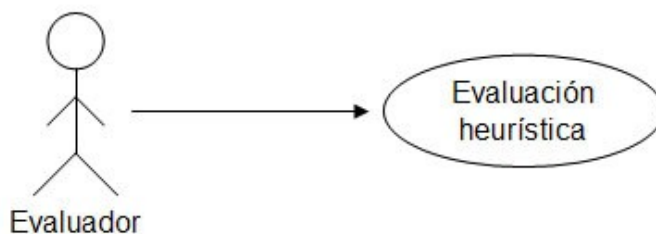


Imagen 26: Caso de uso: Evaluación heurística

Responsable	Sistema
1. El evaluador quiere realiza la evaluación heurística.	
	2. El sistema muestra una lista de los criterios de la heurística.
3. El evaluador imprime una copia de la heurística.	
	4. Se realiza la impresión de la heurística.
5. El evaluador observa la interfaz del proyecto antes de comenzar la evaluación.	
6. Rellena la heurística impresa mientras observa la interfaz.	
7. Pasa los resultados al ordenador y escribe las observaciones necesarias.	
	8. El sistema almacena los resultados de la evaluación.

Tabla 5: Flujo de acontecimientos: Evaluación heurística

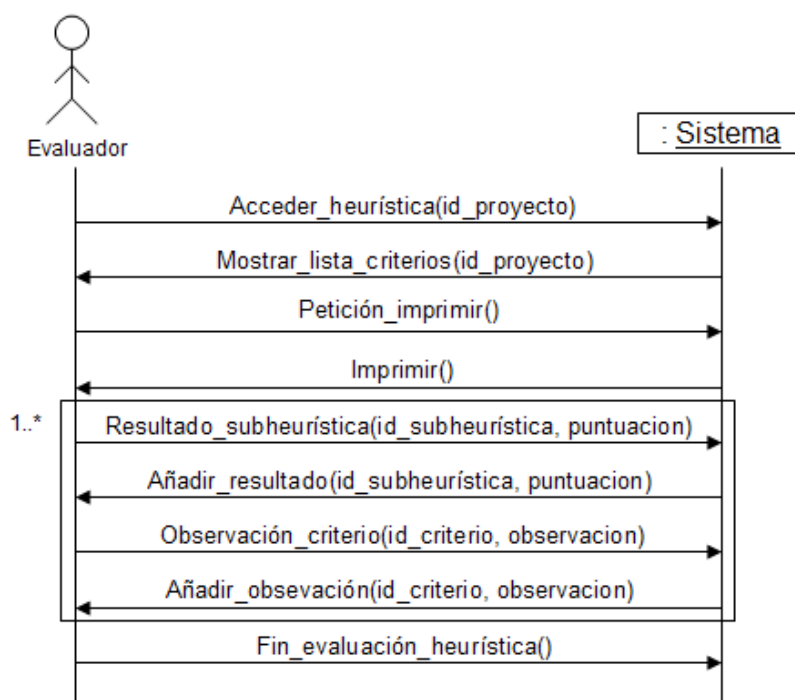


Imagen 27: Diagrama de secuencia: Evaluación heurística

**Pre:** El evaluador ha recibido la heurística.

**Post:** Los resultados y las observaciones de la evaluación se almacenan en el sistema.

**Excepción:** El evaluador no ha recibido la heurística. Los resultados de la heurística ya se han enviado.

### Caso de uso: Envío de los resultados

Una vez realizada la evaluación cada evaluador envía sus resultados al responsable para que realice el análisis global de la evaluación. El sistema indica al responsable los evaluadores que le han enviado los resultados y los que no.

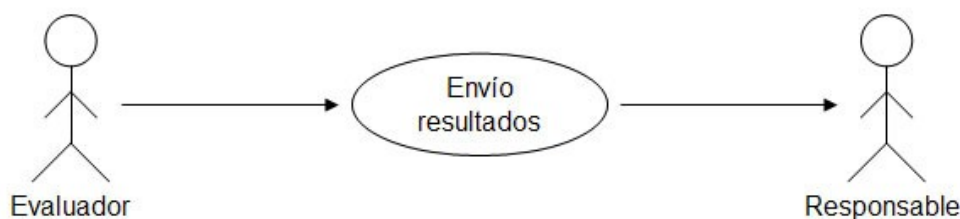


Imagen 28: Caso de uso: Envío de los resultados

Evaluador	Sistema	Responsable
1. El evaluador quiere enviar los resultados de la evaluación.		
	2. El sistema envía los resultados al correo del responsable.	
		3. El responsable recibe la evaluación del evaluador.

Tabla 6: Flujo de acontecimientos: Envío de los resultados

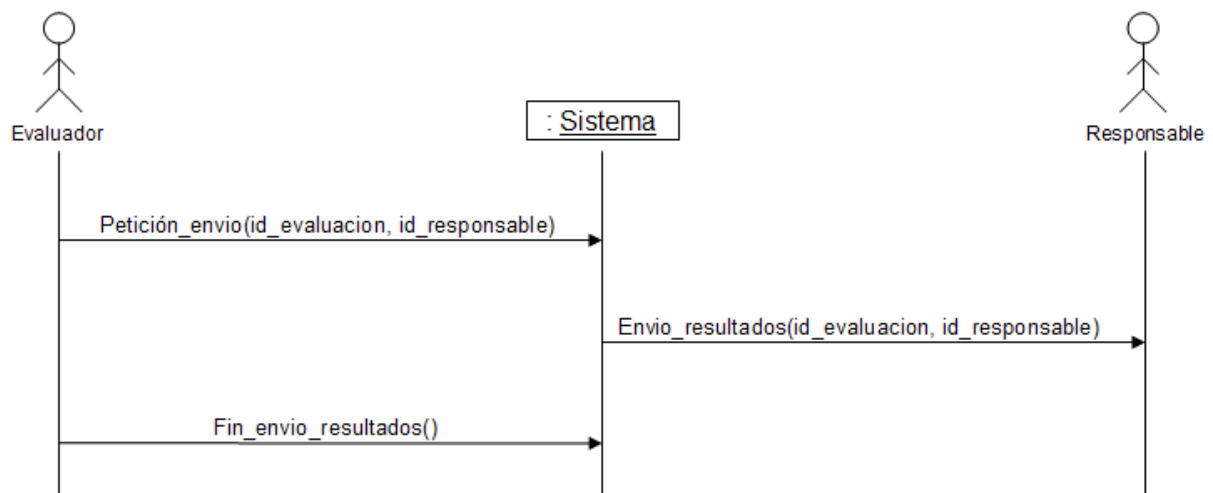


Imagen 29: Diagrama de secuencia: Envío de los resultados

**Pre:** La evaluación heurística ya se ha realizado.

**Post:** El responsable recibe los resultados y puede empezar con el análisis.

**Excepción:** No se ha recibido la evaluación. La evaluación no se ha realizado todavía.

## 4. Análisis de riesgos, planificación y costes

### 4.1. Gestión de riesgos

Los objetivos de la gestión de riesgos son identificar, controlar y eliminar aquello que provoca riesgos antes de que éstos impidan alcanzar los objetivos del proyecto. La gestión de riesgos reduce la probabilidad de fracaso del producto, reduce los ciclos y repeticiones en la producción del software y optimiza los costos y el uso de recursos.

#### 4.1.1. Definición y clasificación de riesgos

El riesgo es una combinación de limitaciones e incertezas que siempre se tienen a la hora de desarrollar un proyecto. Los riesgos aumentan la posibilidad de tener pérdidas en el proyecto como disminución de la calidad, incremento del coste, retraso o fallos.

Las características del riesgo de software son: la incerteza y la pérdida.

- ♦ **Incerteza.** El acontecimiento que provoca el riesgo puede o no ocurrir.
- ♦ **Pérdida.** Si el riesgo se hace realidad, habrá consecuencias no deseadas o pérdidas.

Es necesario cuantificar el nivel de incerteza y el grado de pérdida asociados a cada riesgo. Por eso se dividen los riesgos en diferentes categorías.

- ♦ **Riesgos del proyecto.** Afectan a la planificación temporal, al coste y a la calidad del proyecto.
- ♦ **Riesgo técnico.** Afectan a la calidad y a la planificación temporal del software en desarrollo.
- ♦ **Riesgo del negocio.** Amenazan la viabilidad del software en el mercado.

Los riesgos también se pueden clasificar según su facilidad de detección [Press01]:

- ♦ **Riesgos conocidos.** Se pueden predecir después de una evaluación del plan de proyecto, del entorno técnico y otras fuentes de información fiables.
- ♦ **Riesgos predecibles.** Se aplican de la experiencia de proyectos anteriores.
- ♦ **Riesgos impredecibles.** Son aquellos difíciles de identificar por adelantado.

Para conseguir una buena gestión de riesgos, ésta debe ser continuada. Se debe evaluar continuamente aquellos eventos que pueden fallar, detectar los riesgos más importantes y desarrollar estrategias eficaces para solucionarlos.

Existen dos estrategias principales para el tratamiento de los riesgos: la reactiva y la proactiva.

- ◆ **Reactiva.** Pensar en los problemas en el momento en que éstos suceden.
- ◆ **Proactiva.** Se identifican los riesgos potenciales, se valora la probabilidad, su impacto y la prioridad.

#### 4.1.2. Estimación del riesgo

##### Identificación del riesgo

Para identificar los riesgos es necesario realizar una lista de riesgos potenciales que pueden afectar a la planificación del proyecto.

Pueden existir dos tipos de riesgos: específicos y generales.

- ◆ **Específicos.** Están relacionados con la tecnología, el personal y el entorno específico. Para identificarlos es necesario examinar el plan del proyecto y los requisitos del software.
- ◆ **Generales.** Son una amenaza potencial para todos los proyectos. Para identificarlos se crean diferentes subcategorías. Junto a las subcategorías se describen los riesgos que pueden afectar al desarrollo de este proyecto:
  - **Tamaño del producto.** En este proyecto se tiene que tener en cuenta el tamaño del documento XML que se utilizará como base de datos y el número de documentos de heurísticas y evaluaciones que podrá requerir la aplicación. Hay que tener también en cuenta que el cliente puede cambiar los requisitos del sistema o ampliar el proyecto.
  - **Impacto en el negocio.** El riesgo principal que se puede producir en este caso es que se tenga que aplazar la fecha de entrega del proyecto final.
  - **Características del cliente.**
  - **Definición del proceso.**
  - **Entorno de desarrollo.** Es importante saber las limitaciones y prestaciones del software de desarrollo y comprobar que las tecnologías empleadas son compatibles.
  - **Tamaño y experiencia de la plantilla.** Los miembros de la plantilla pueden no tener conocimientos sobre las tecnologías de desarrollo empleadas.

## Análisis de riesgos

En el análisis de riesgos se mide la probabilidad y el impacto de cada riesgo. Se examinan los riesgos con detalle para conocer su importancia, su extensión y su impacto.

Con los riesgos identificados anteriormente se realiza una tabla determinando la probabilidad de que se produzca el riesgo y el impacto de cada uno. El impacto se define por: catastrófico, crítico, marginal o despreciable.

Riesgos	Categoría	Probabilidad	Impacto
Tamaño y número de los documentos	Tamaño del proyecto	30%	Marginal
Cambio o ampliación del proyecto	Tamaño del proyecto	60%	Crítico
No cumplir con la fecha de entrega	Impacto en el negocio	80%	Crítico
Limitaciones del software	Entorno de desarrollo	50%	Crítico
No conocimiento de las tecnologías empleadas	Experiencia de la plantilla	70%	Marginal

*Tabla 7: Tabla de riesgos del proyecto*

### 4.1.3. Control de riesgos

Una vez identificados y analizados los posibles riesgos del proyecto se debe realizar una reducción, una supervisión y una gestión de los mismos en caso de que se produzcan. Este control sirve para evitar que se produzcan los riesgos antes de comenzar el proyecto, para evitar que surjan durante el desarrollo del proyecto y para disminuir las consecuencias si los riesgos se producen.

A continuación se aplican las tareas de reducción, supervisión y gestión a cada riesgo identificado.

#### ♦ Tamaño y número de los documentos

- Reducción: se evitará que haya documentos iguales almacenados. Los documentos con el mismo nombre se sobrescribirán y todos los documentos que ya no se puedan emplear serán eliminados.
- Supervisión: se revisará que el número de documentos utilizados sea más o menos constante.
- Gestión: replantear el diseño de base de datos.



- ◆ **Cambio o ampliación del proyecto**

- Reducción: el cliente tendrá que especificar claramente lo que quiere antes del comienzo del proyecto y confirmar que se ha entendido lo que pide.
- Supervisión: reuniones continuas para asegurar que los requisitos del sistema son correctos.
- Gestión: intentar que el cambio o ampliación afecte lo menos posible.

- ◆ **No cumplir con la fecha de entrega**

- Reducción: realizar una planificación temporal del proyecto especificando la duración de cada tarea para poder fijar una fecha razonable.
- Supervisión: dependerá principalmente de las modificaciones o imprevistos que surjan durante el desarrollo del proyecto.
- Gestión: intentar encontrar una fecha lo más próxima posible.

- ◆ **Limitaciones del software**

- Reducción: realizar un estudio sobre las herramientas necesarias para el desarrollo del proyecto.
- Supervisión: conforme se desarrolle el proyecto se podrán tomar las decisiones necesarias para no tener limitaciones de software.
- Gestión: encontrar otra herramienta similar que no restrinja el proyecto.

- ◆ **No conocimiento de las tecnologías empleadas**

- Reducción: antes de comenzar a trabajar con una herramienta, estudiar su funcionamiento.
- Supervisión: si hay alguna duda, preguntar a expertos o buscar información adicional.
- Gestión: estudiar manuales sobre la herramienta.

## 4.2. Planificación del proyecto

La planificación del proyecto proporciona un marco de trabajo que permita hacer estimaciones razonables de recursos, costes y planificación temporal del proyecto. La planificación se realiza al inicio del proyecto, pero es necesario modificarla a medida que el proyecto avanza.

Para hacer la planificación se han de identificar todas las tareas que se tienen que realizar a lo largo del proyecto, determinar su duración, su relación de precedencia y los recursos a utilizar.

La planificación inicial del proyecto era:

<b>Fases del proyecto</b>	<b>Periodo</b>
Análisis de requisitos	Octubre 2006 – Diciembre 2006
Diseño	Octubre 2006 – Noviembre 2006
Prototipos	Noviembre 2006 – Diciembre 2006
Implementación	Diciembre 2006 – Marzo 2007
Evaluación	Marzo 2007 – Abril 2007
Documentación	Noviembre 2006 – Abril 2007

*Tabla 8: Planificación inicial del proyecto*

Las primeras fases del proyecto siguieron la planificación inicial que se había especificado, pero una vez en la fase de implementación se tuvo que modificar la planificación temporal debido a los imprevistos ocasionados. En primer lugar, la implementación del sistema se hizo utilizando como base de datos Microsoft Access, pero a lo largo del desarrollo de la aplicación se optó por una base de datos basada en documentos XML utilizando la tecnología DOM. Este cambio se realizó porque si un usuario de la aplicación no tenía instalado Microsoft Access en su ordenador, la aplicación no funcionaría. Además de tener que modificar el código de implementación, el programador del sistema no tenía experiencia con la tecnología DOM y XML integrado en Visual Basic, por lo que la duración de la fase de implementación aumentó. En segundo lugar, llegados a un punto de la implementación, la versión utilizada de Visual Basic .NET no admitía algunas funciones imprescindibles de la interfaz del DOM, teniendo que buscar otra versión más reciente. Todos estos imprevistos fueron identificados en la gestión de riesgos del proyecto.

Finalmente la planificación temporal del proyecto es:

<b>Fases del proyecto</b>	<b>Periodo</b>
Análisis de requisitos	Octubre 2006 – Abril 2007
Diseño	Octubre 2006 – Noviembre 2006
Prototipos	Noviembre 2006 – Diciembre 2006
Implementación	Diciembre 2006 – Septiembre 2007
Evaluación	Julio 2007 – Septiembre 2007
Documentación	Noviembre 2006 – Septiembre 2007

*Tabla 9: Planificación final del proyecto*

### 4.3.Presupuesto

El presupuesto es el coste aproximado de todo el proyecto que se le va a proponer al cliente. Para calcular los costes de un proyecto el presupuesto se divide en:

- ♦ **Inventariable.** Son los bienes materiales que se utilizan durante el desarrollo del proyecto como el equipo informático o las aplicaciones de desarrollo. Como estos materiales se pueden utilizar en más de un proyecto, en vez de contar el precio total del producto, se cuenta con la amortización de éste durante el periodo que se ha utilizado. El hardware puede tener una amortización de hasta 4 años, pero en este proyecto se ha utilizado durante 12 meses. Respecto al software, los sistemas operativos tienen una duración de hasta 4 años y las aplicaciones empleadas para el desarrollo del proyecto tienen una duración más corta debido a sus constantes actualizaciones y hay que tener en cuenta las aplicaciones y tecnologías de libre distribución.

Material	Amortización	Precio
Equipo informático	1100€ / 48meses(vida) * 12meses(útil)	275€
Microsoft Windows XP Professional	120€ / 48meses(vida) * 12meses(útil)	30€
Microsoft Visual Studio .NET Professional	730€ / 24meses(vida) * 10meses(útil)	305€
Microsoft Office XP	220€ / 48meses(vida) * 12meses(útil)	55€
OpenOffice 2.0		0€
Adobe Photoshop	600€ / 24meses(vida) * 2meses(útil)	50€
Total		715€

Tabla 10: Coste Inventariable

- ♦ **Fungible.** Son los materiales que se consumen o se agotan como papel, cd's, tinta de imprimir, etc.

Material	Precio/Unidad	Cantidad	Precio
Lata 10 CD's	5,60€	1	5,60€
Paquete de folios de 500 hojas	6,25€	1	6,25€
Tinta negra	17€	1	17€
Tinta color	29€	1	29€
Total			57,85€

Tabla 11: Coste Fungible

- ♦ **Personal.** Es el salario de las personas que participan en el desarrollo del proyecto. Se ha considerado un coste de 15€ por hora de trabajo.

Material	Precio/Unidad	Cantidad	Precio
Programador	15€/hora	1.000 horas	15.000€
Total			15.000€

Tabla 12: Coste Personal

- ♦ **Viajes y dietas.** Son los gastos de transporte y comidas que se hayan producido a causa de las reuniones.

Material	Precio/Unidad	Cantidad	Precio
Desplazamientos	30€	6	180€
Total			180€

Tabla 13: Coste Viajes y dietas

- ♦ **Overhead.** Es un porcentaje que se aplica sobre el precio total del proyecto para considerar los gastos indirectos y los imprevistos. Se ha considerado un overhead del 10%.
- ♦ **Beneficio.** Es un porcentaje que se aplica sobre el precio total del proyecto para sacar un beneficio de la producción del proyecto. Se ha considerado un beneficio del 15%.

El presupuesto total del proyecto según los costes calculados es:

Inventariable	715€
Fungible	57,85€
Personal	15.000€
Viajes y dietas	180€
SUBTOTAL	15.952,85€
Overhead	1.595,28€
SUBTOTAL	17.548,13€
Beneficio	2.632,22€
<b>TOTAL</b>	<b>20.180,35€</b>

Tabla 14: Presupuesto final

## 5. Diseño

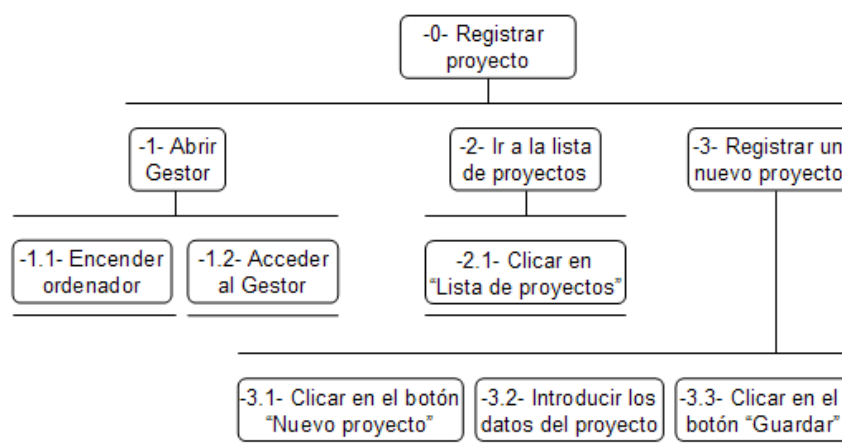
### 5.1. Análisis jerárquico de tareas

El análisis jerárquico de tareas es un estudio de las acciones que tendrán que realizar los usuarios del sistema para alcanzar sus objetivos. Facilita la comprensión del sistema y de los flujos de información.

Las tareas vienen representadas en forma de diagrama de árbol y cada hoja del diagrama representa una acción de la tarea.

A continuación se detalla el análisis jerárquico de tareas de este proyecto.

- ♦ Registrar un proyecto



Plan 0: Hacer 1 si la aplicación está cerrada o 2 si se está dentro de un proyecto y hacer 3.

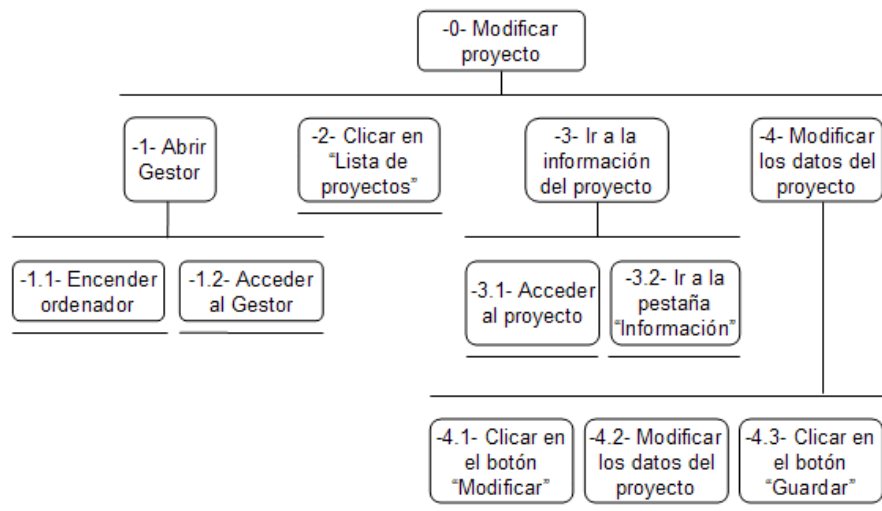
Plan 1: Hacer 1.1 y 1.2.

Plan 2: Hacer 2.1.

Plan 3: Hacer 3.1, 3.2 y 3.3

*Imagen 30: Tarea: Registrar proyecto*

♦ Modificar un proyecto



Plan 0: Hacer 1 si la aplicación está cerrada o 2 si se está dentro de otro proyecto y hacer 3 y 4.

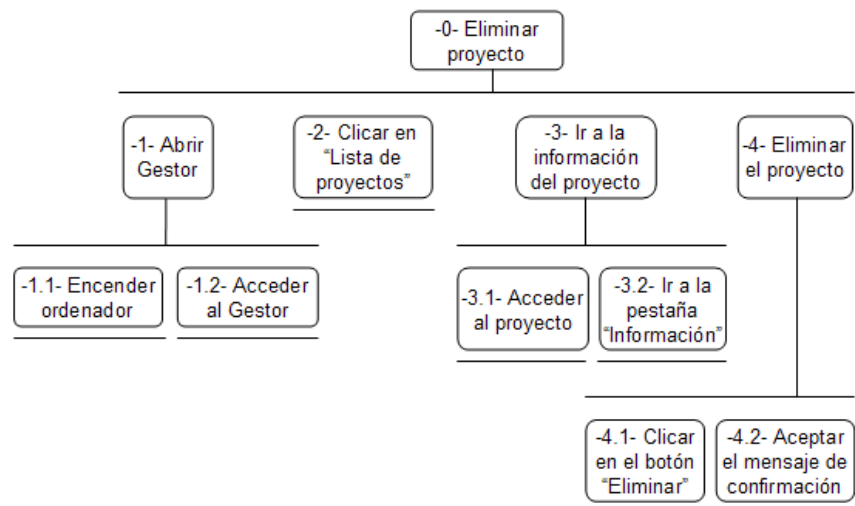
Plan 1: Hacer 1.1 y 1.2.

Plan 3: Hacer 3.1 y 3.2.

Plan 4: Hacer 4.1, 4.2 y 4.3.

*Imagen 31: Tarea: Modificar proyecto*

♦ Eliminar un proyecto



Plan 0: Hacer 1 si la aplicación está cerrada o 2 si se está dentro de otro proyecto y hacer 3 y 4.

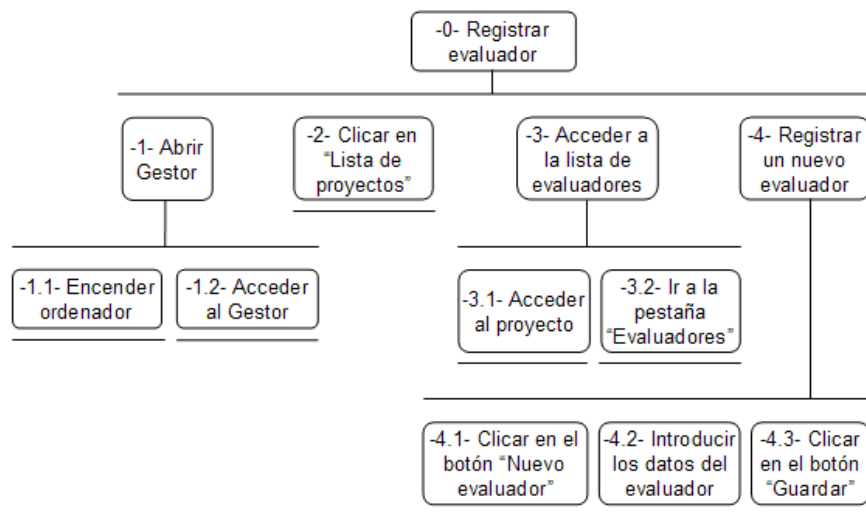
Plan 1: Hacer 1.1 y 1.2.

Plan 3: Hacer 3.1 y 3.2.

Plan 4: Hacer 4.1 y 4.2.

*Imagen 32: Tarea: Eliminar proyecto*

◆ Registrar un evaluador



Plan 0: Hacer 1 si la aplicación está cerrada o 2 si se está dentro de otro proyecto y hacer 3 y 4.

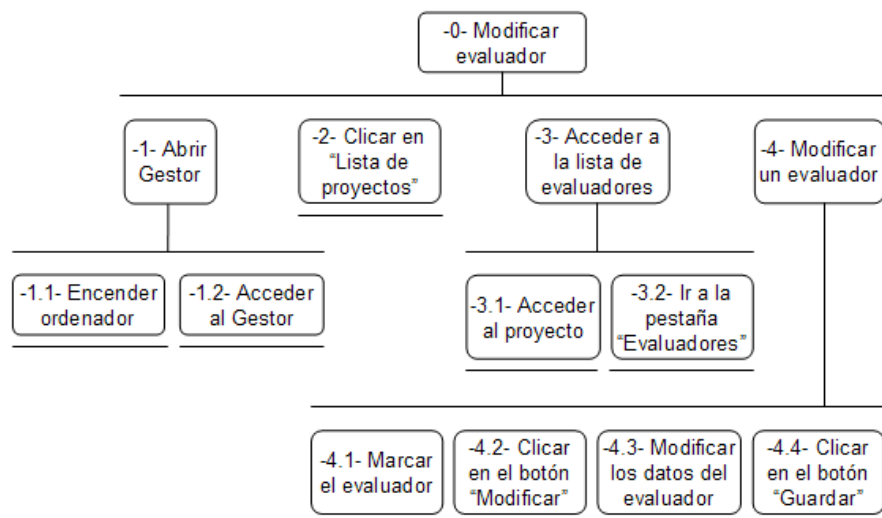
Plan 1: Hacer 1.1 y 1.2.

Plan 3: Hacer 3.1 y 3.2.

Plan 4: Hacer 4.1, 4.2 y 4.3.

*Imagen 33: Tarea: Registrar evaluador*

◆ Modificar un evaluador



Plan 0: Hacer 1 si la aplicación está cerrada o 2 si se está dentro de otro proyecto y hacer 3 y 4.

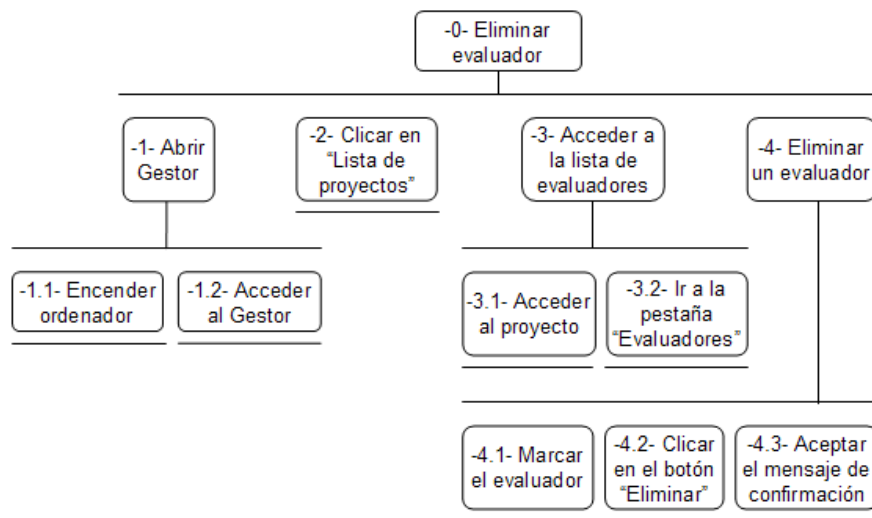
Plan 1: Hacer 1.1 y 1.2.

Plan 3: Hacer 3.1 y 3.2.

Plan 4: Hacer 4.1, 4.2, 4.3 y 4.4.

*Imagen 34: Tarea: Modificar evaluador*

◆ Eliminar un evaluador



Plan 0: Hacer 1 si la aplicación está cerrada o 2 si se está dentro de otro proyecto y hacer 3 y 4.

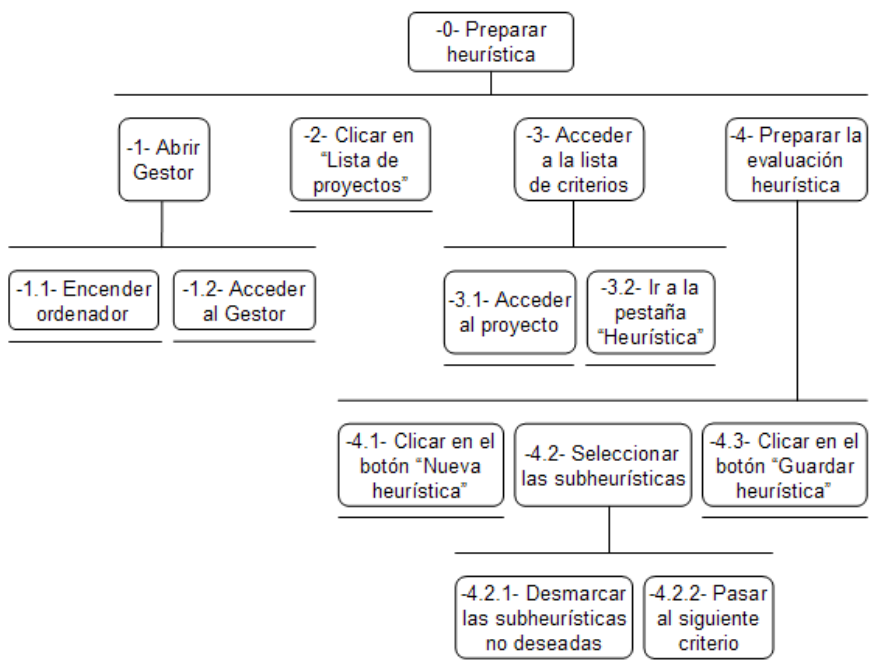
Plan 1: Hacer 1.1 y 1.2.

Plan 3: Hacer 3.1 y 3.2.

Plan 4: Hacer 4.1, 4.2 y 4.3.

*Imagen 35: Tarea: Eliminar evaluador*

◆ Preparar la heurística



Plan 0: Hacer 1 si la aplicación está cerrada o 2 si se está dentro de otro proyecto y hacer 3 y 4.

Plan 1: Hacer 1.1 y 1.2.

Plan 3: Hacer 3.1 y 3.2.

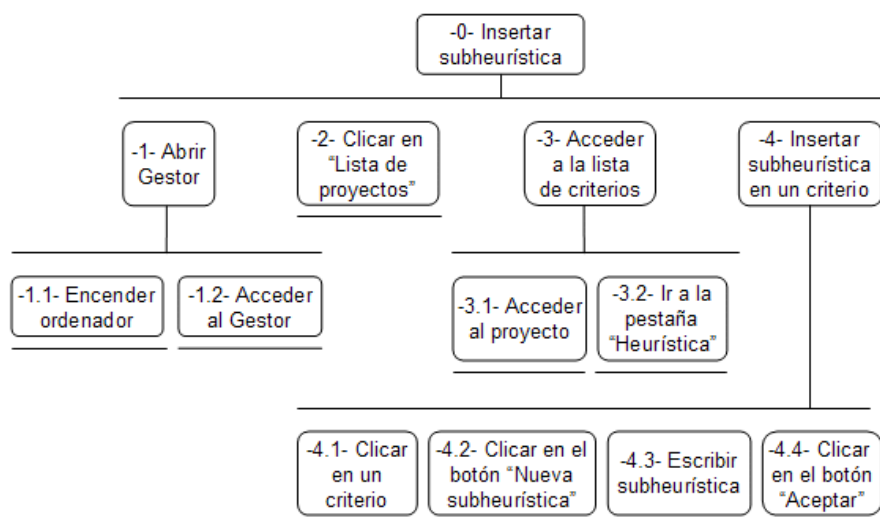
Plan 4: Hacer 4.1, 4.2 y 4.3.

Plan 4.2: Hacer 4.2.1 y 4.2.2 las veces necesarias.

*Imagen 36: Tarea: Nueva heurística*



◆ Insertar subheurísticas nuevas



Plan 0: Hacer 1 si la aplicación está cerrada o 2 si se está dentro de otro proyecto y hacer 3 y 4.

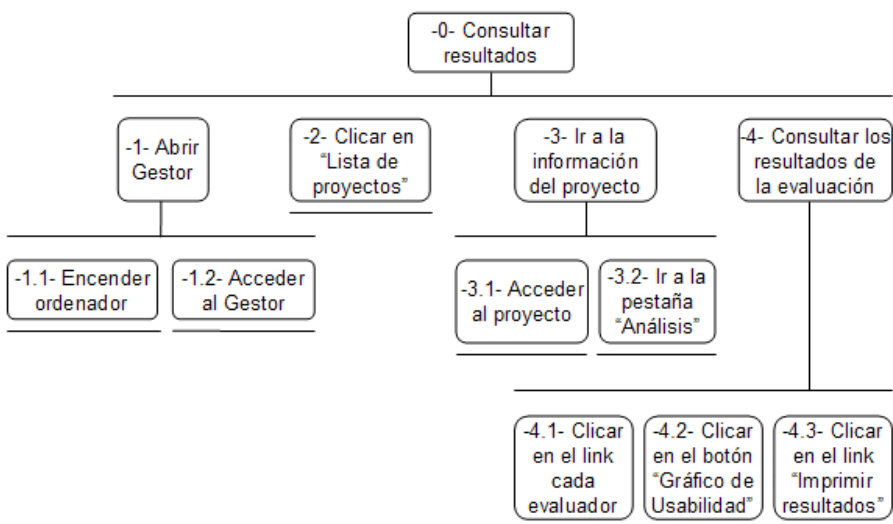
Plan 1: Hacer 1.1 y 1.2.

Plan 3: Hacer 3.1 y 3.2.

Plan 4: Hacer 4.1, 4.2, 4.3 y 4.4.

*Imagen 37: Tarea: Insertar subheurística*

◆ Consultar los resultados de la evaluación



Plan 0: Hacer 1 si la aplicación está cerrada o 2 si se está dentro de otro proyecto y hacer 3 y 4.

Plan 1: Hacer 1.1 y 1.2.

Plan 3: Hacer 3.1 y 3.2.

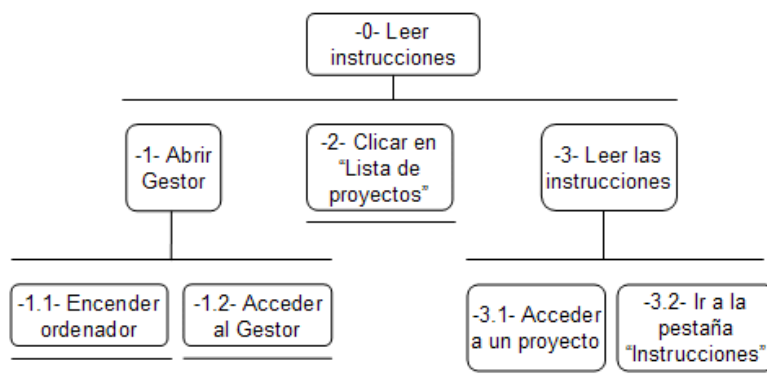
Plan 4: Hacer 4.1 para ver los resultados de cada evaluador.

Hacer 4.2 para ver el gráfico de los resultados finales de la evaluación.

Hacer 4.3 para imprimir los resultados de la evaluación.

*Imagen 38: Tarea: Consultar resultados*

- ◆ Leer las instrucciones de la evaluación



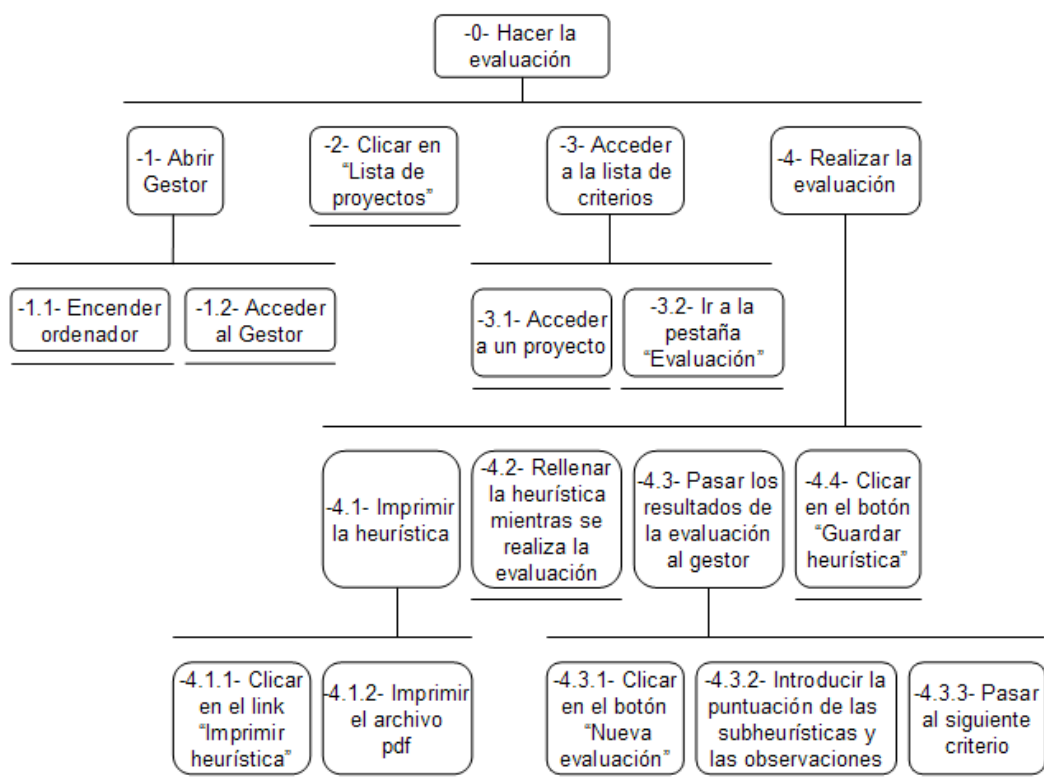
Plan 0: Hacer 1 si la aplicación está cerrada o 2 si se está dentro de otro proyecto y hacer 3.

Plan 1: Hacer 1.1 y 1.2.

Plan 3: Hacer 3.1 y 3.2.

*Imagen 39: Tarea: Leer instrucciones*

- ◆ Realizar la evaluación del sistema



Plan 0: Hacer 1 si la aplicación está cerrada o 2 si se está dentro de otro proyecto y hacer 3 y 4.

Plan 1: Hacer 1.1 y 1.2.

Plan 3: Hacer 3.1 y 3.2.

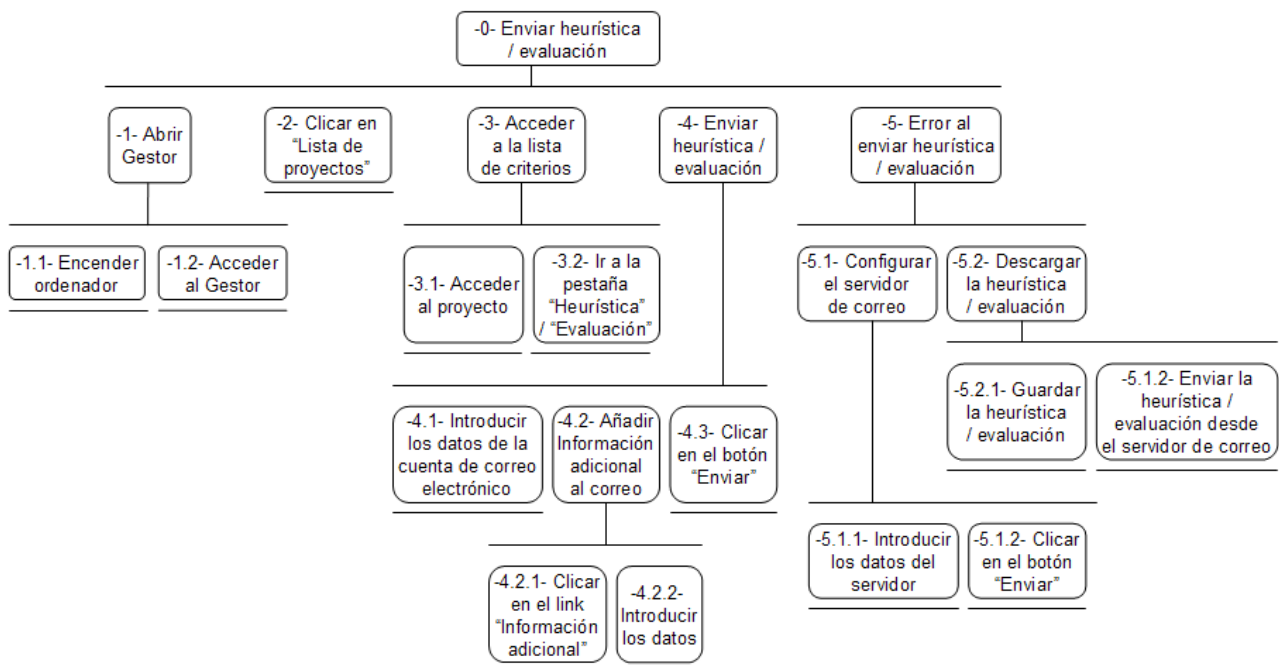
Plan 4: Hacer 4.1, 4.2, 4.3 y 4.4.

Plan 4.1: Hacer 4.1.1 y 4.1.2.

Plan 4.3: Hacer 4.3.1, 4.3.2 y 4.3.3.

*Imagen 40: Tarea: Hacer evaluación*

◆ Enviar la heurística (caso Responsable) / Enviar la evaluación (caso Evaluador)



Plan 0: Hacer 1 si la aplicación está cerrada o 2 si se está dentro de otro proyecto, hacer 3 y 4 y hacer 5 si se produce un error al enviar.

Plan 1: Hacer 1.1 y 1.2.

Plan 3: Hacer 3.1 y 3.2.

Plan 4: Hacer 4.1, 4.2 si se quiere añadir información adicional al correo y hacer 4.3.

Plan 4.2: Hacer 4.2.1 y 4.2.2.

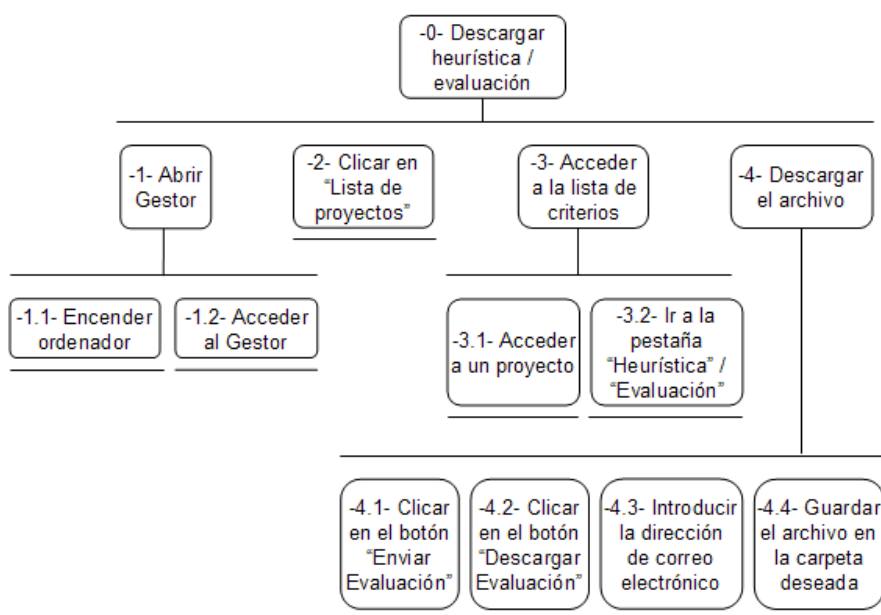
Plan 5: Hacer 5.1 si se quiere configurar el servidor o hacer 5.2 si se quiere descargar la heurística/evaluación.

Plan 5.1: Hacer 5.1.1 y 5.1.2.

Plan 5.2: Hacer 5.2.1 y 5.2.2.

*Imagen 41: Tarea: Enviar heurística/evaluación*

- ◆ Descargar la heurística (caso Responsable) / Descargar la evaluación (caso Evaluador)



Plan 0: Hacer 1 si la aplicación está cerrada o 2 si se está dentro de otro proyecto y hacer 3 y 4.

Plan 1: Hacer 1.1 y 1.2.

Plan 3: Hacer 3.1 y 3.2.

Plan 4: Hacer 4.1, 4.2 y 4.3.

*Imagen 42: Tarea: Descargar heurística/evaluación*

## 5.2. Interfaz gráfica

En el diseño de la interfaz gráfica se ha seguido el mismo modelo en los dos módulos que componen la aplicación.

La aplicación tiene que ser fácil de entender y aprender. Por este motivo se han organizado todas las actividades que constituyen la evaluación heurística de manera separada.

En ambos módulos cuando se ejecuta la aplicación aparece una lista con los proyectos que contiene el gestor. Una vez dentro de un proyecto, las tareas a llevar a cabo se organizan por pestañas.

En el módulo del Responsable, la primera pestaña contiene toda la información relevante al proyecto y el punto en el que se encuentra la evaluación. En la segunda pestaña se muestra una lista de los evaluadores elegidos para realizar la evaluación heurística junto con la información de éstos. La tercera pestaña contiene todas las actividades relacionadas con la heurística como son: crear heurística, insertar nueva subheurística y enviar heurística. Y en la cuarta pestaña se realiza el análisis de la evaluación con los resultados recibidos de los evaluadores.

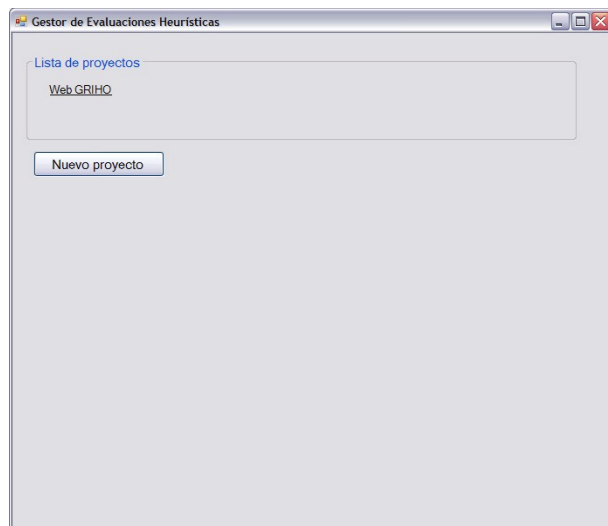


Imagen 43: Lista de proyectos

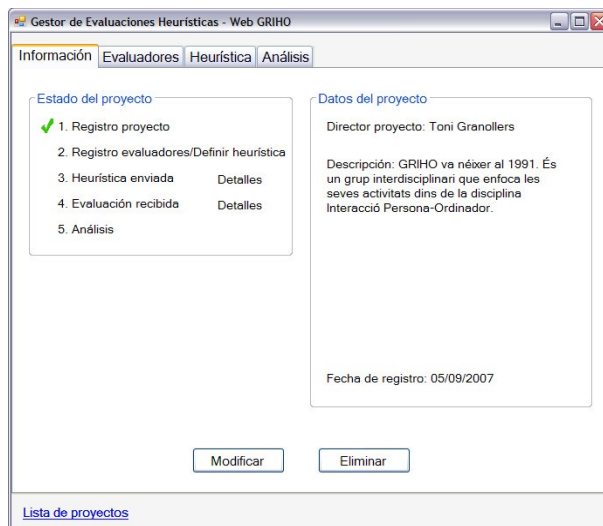


Imagen 44: Pestaña Información

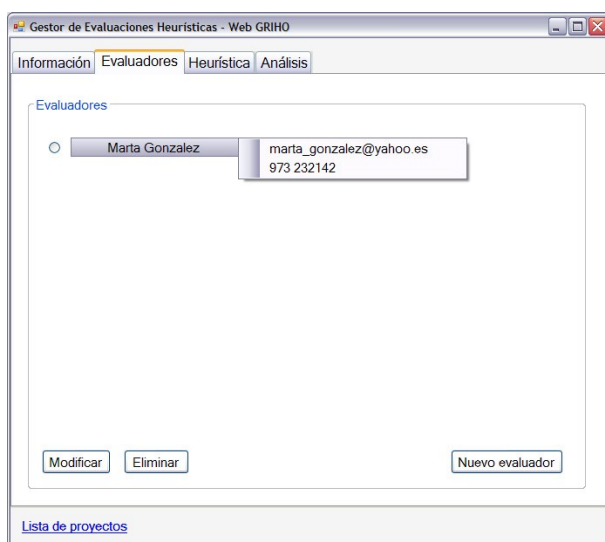


Imagen 45: Pestaña Evaluadores

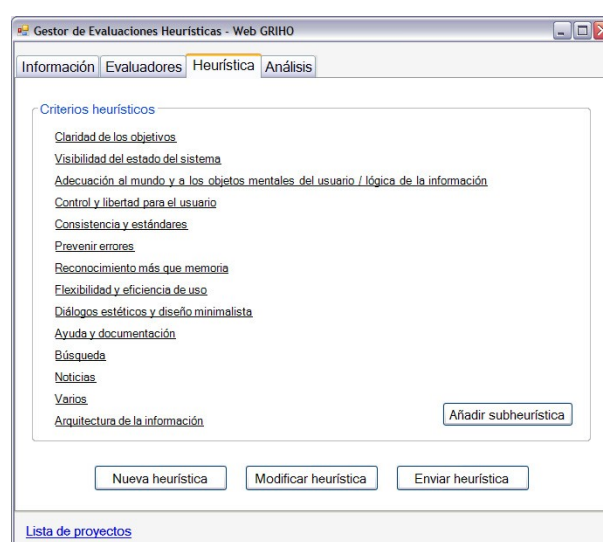


Imagen 46: Pestaña Heurística



Imagen 47: Pestaña Análisis

En el módulo del Evaluador, la primera pestaña contiene la información del proyecto que le servirá al evaluador para conocer un poco más el proyecto que tiene que evaluar. En la segunda pestaña están las instrucciones de cómo el evaluador tiene que realizar la evaluación heurística. Y la tercera pestaña contiene las tareas relacionadas con la evaluación: crear evaluación, imprimir heurística y enviar evaluación.

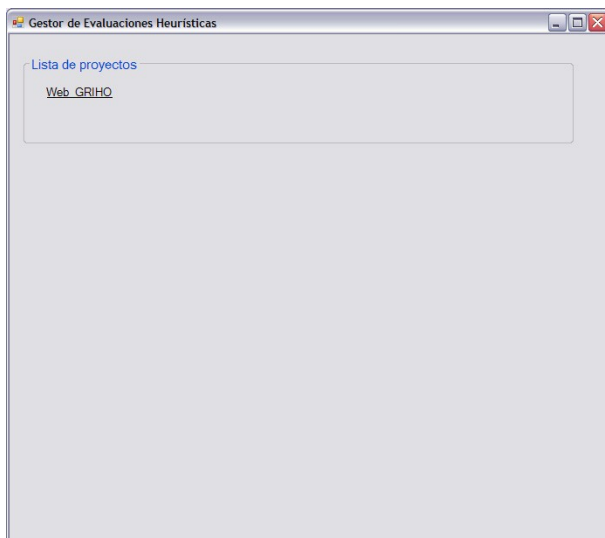


Imagen 48: Lista de proyectos

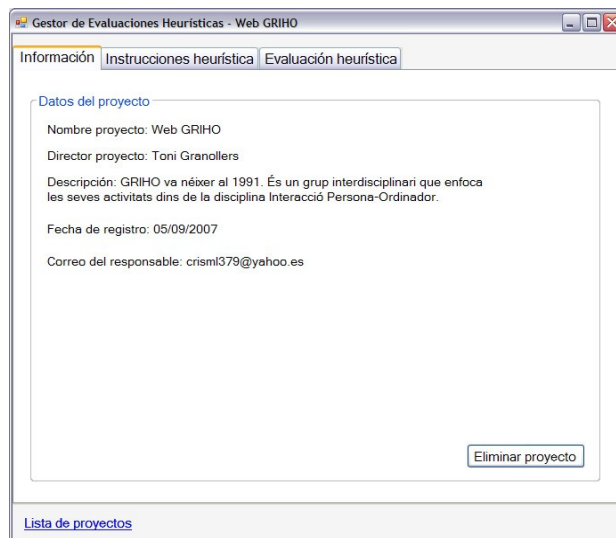


Imagen 49: Pestaña Información

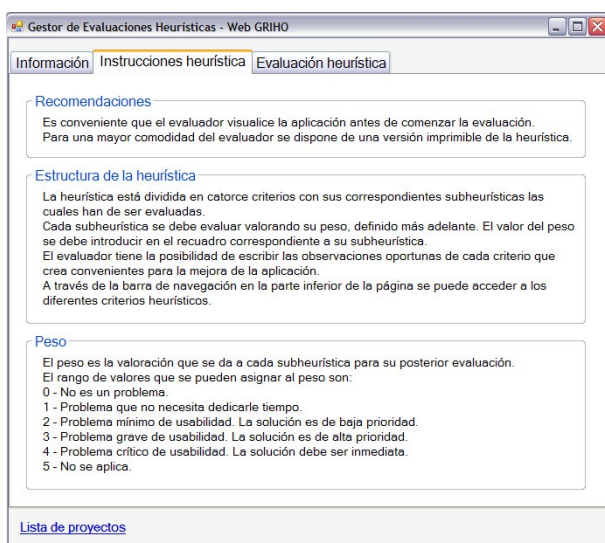


Imagen 50: Pestaña Instrucciones

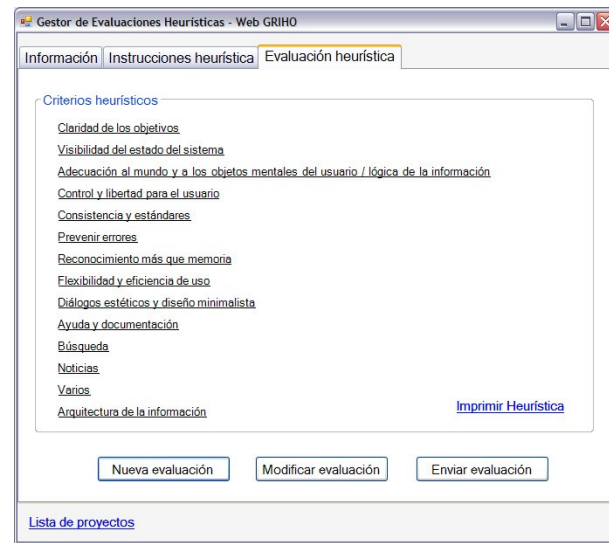


Imagen 51: Pestaña Evaluación

## 6. Implementación

Para el desarrollo e implementación del sistema se han utilizado diferentes tecnologías. Para la implementación del código se ha utilizado la tecnología .NET mediante la plataforma de desarrollo Microsoft Visual Studio .NET y para administrar la información del sistema se ha utilizado la tecnología XML.

### 6.1. Plataforma .NET

#### 6.1.1. ¿Qué es .NET?

.NET es una nueva tecnología desarrollada por Microsoft para la creación y distribución de software. Mediante las herramientas de desarrollo proporcionadas para esta tecnología, los programadores podrán crear aplicaciones y permite que los sistemas y aplicaciones conecten sus datos y transacciones independientemente del sistema operativo, tipo de computadora o dispositivo móvil que se utilice, o del lenguaje de programación empleados para crearlo.

Las características principales que establecen la tecnología .NET son:

- ♦ **La plataforma .NET Framework.** Proporciona la infraestructura adecuada para crear aplicaciones y el entorno de ejecución de éstas.
- ♦ **Los productos de Microsoft enfocados hacia .NET.** Como sistema operativo está Windows .NET Server y como herramienta de desarrollo de las aplicaciones Visual Studio .NET.
- ♦ **Servicio para .NET desarrollados por terceros fabricantes.**

#### 6.1.2. .NET Framework

.NET Framework constituye la plataforma y elemento principal de Microsoft .NET, proporcionando las herramientas y servicios necesarios para el desarrollo de aplicaciones. Las herramientas y servicios proporcionados pueden agruparse en tres bloques: el Entorno de Ejecución Común (Common Language Runtime, CLR), la jerarquía de clases básicas de la plataforma y el motor de generación de interfaz de usuario.

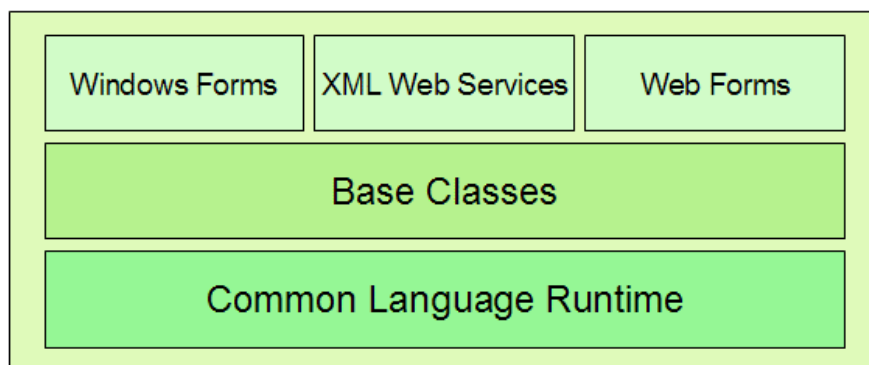


Imagen 52: Esquema de componentes dentro de la plataforma .NET Framework

- ♦ **CLR.** Se encuentra en la base del entorno de ejecución y constituye el núcleo de .NET Framework. El CLR se encarga de la gestión del código en cuanto a su carga, ejecución, manipulación de memoria, seguridad, etc.  
Las características principales del CLR son:
  - Proporciona un desarrollo de aplicaciones más sencillo y rápido, ya que gran parte de las funcionalidades vienen implementadas en el entorno de ejecución.
  - Administra el código en tiempo de ejecución.
  - Implementa características de gestión a bajo nivel.
  - Proporciona un sistema común de tipo para todos los lenguajes del entorno (CLS).
  - Gestiona la seguridad del código que es ejecutado.
  - Dispone de un diseño abierto a lenguajes y herramientas de desarrollo creadas por otros fabricantes.
  - Facilita la distribución e instalación de aplicaciones, ya que es posible instalar una aplicación simplemente copiando los ficheros que la componen en uno de los directorios del equipo en el que se vaya a ejecutar.
- ♦ **Jerarquía de clases básicas del entorno de ejecución.** Se sitúa en el nivel intermedio y constituye una interfaz de programación de aplicaciones (API) de servicios a disposición del programador para multitud de tareas. La mayor parte de las necesidades básicas del programador están cubiertas por este amplio conjunto de clases, que permiten dotar a las aplicaciones de todas las características necesarias y cuya jerarquía está diseñada en torno a la programación orientada a objetos con una única clase raíz de la que derivan todas las demás clases.



- ♦ **Motor de generación de interfaz de usuario.** Se encuentra en el nivel superior y permite crear interfaces para las aplicaciones. La plataforma pone a disposición de los programadores para el sistema operativo Windows, una nueva generación de formularios con características avanzadas llamados Windows Forms.

[Bla05]

### 6.1.3. Visual Basic .NET

Microsoft Visual Studio .NET es un entorno de desarrollo integrado (IDE) para aplicaciones basadas en la plataforma .NET de Microsoft. Con esta herramienta se pueden crear aplicaciones de escritorio, para Internet y para dispositivos móviles y cargar otros tipos de archivos y utilizarla como editor de recursos, editor de imágenes, explorador de HTML, etc.

Este IDE utiliza diferentes tipos de lenguaje para el desarrollo de aplicaciones: Visual Basic, C#, JScript y C++.

Visual Basic .NET es un lenguaje específico para aplicaciones basadas en .NET. Este lenguaje puede cooperar sin limitaciones con otros lenguajes de la plataforma gracias al CLS de .NET Framework. Con este lenguaje se pueden desarrollar diferentes tipos de proyectos: componentes, bibliotecas de clases, aplicaciones y servicios Web XML, entre otros, pero los proyectos más importantes son aplicaciones y servicios para Windows. Estas aplicaciones pueden ser tanto de escritorio como para Internet y Servicios Web XML basados en .NET Framework.

Dentro de los proyectos este lenguaje ofrece la posibilidad de definir ámbitos con nombre, clases que pueden derivarse de otras mediante herencia, sobrecarga de métodos, control estructurado de excepciones o la creación de aplicaciones con múltiples hilos de ejecución, etc.

## 6.2. XML y DOM

### 6.2.1. XML y DTD

XML (eXtensible Markup Language) es un metalenguaje extensible de etiquetas desarrollado por World Wide Web Consortium (W3C) que permite definir lenguajes de marcado mediante la definición de etiquetas personalizadas para describir y organizar datos. El XML se propone como un estándar de intercambio de información estructurada entre diferentes plataformas, de forma sencilla, segura y fiable, estando presente en sistemas de bases de datos, editores de texto, hojas de cálculos, etc.

XML se basa en tres pilares básicos:

- ♦ **Extensibilidad.** XML acepta cualquier dato que pueda ser descrito como texto y pueda usar etiquetas XML. El lenguaje XML tiene una sintaxis básica determinada y sus elementos y especificaciones se definen en la definición del esquema XML (XSD) o en

la definición de tipo de documento (DTD). Estos documentos contienen las definiciones de los elementos del archivo XML, como están organizados y relacionados y los atributos y tipo que pueden contener.

- ♦ **Estructura.** La estructura de XML es difícil y compleja. Para leer su contenido se utilizan analizadores gramaticales y otros tipos de herramientas diseñadas para trabajar con XML fácilmente. XML fue diseñado para ser un formato de intercambio de datos abierto, y las representaciones de datos con él son por lo general mucho más grandes que sus formatos originales, pero los textos basados en formato XML se comprimen muy bien utilizando la compresión de datos regular y utilidades de transporte.
- ♦ **Validación.** Los datos representados con XML pueden ser validados tanto en la estructura como en el contenido, basándose en dos normas de validación de datos separadas. El estándar XML original de validación de datos se llama DTD (Definición de Tipo de Datos), y la evolución más reciente de validación de datos XML es el estándar de Esquema XML (XSD).

La utilización de XML aporta numerosas ventajas a las aplicaciones donde se utiliza. Algunas de estas ventajas son:

- ♦ Es fácil de procesar y entender.
- ♦ Puede ser utilizado por cualquier lenguaje de programación.
- ♦ Separa el contenido y el formato de presentación.
- ♦ Facilita la comunicación de datos entre aplicaciones.
- ♦ Es extensible, por lo que una vez diseñado el lenguaje y puesto en producción se puede extender añadiendo nuevas etiquetas.
- ♦ Permite fuertes técnicas de extracción y búsqueda de información.

Como se ha explicado anteriormente, se han de establecer unas reglas para la construcción de documentos XML. En la aplicación que se desarrolla para este proyecto se utilizan DTD's para la validación de datos.

Una DTD (Definición de Tipos de Datos) es un conjunto de reglas que definen un documento XML y que permiten etiquetarlo adecuadamente. En una DTD se definen los elementos de un documento XML y como éstos se organizan y relacionan. Cuando un documento XML se ajusta a su DTD se denomina válido.

Las ventajas de utilizar una DTD son:

- ♦ Valida el documento XML.
- ♦ Ofrece fiabilidad y ahorra gran trabajo de análisis y de programación.

- ◆ Especifica la estructura del documento XML.
- ◆ Integridad referencial mínima.
- ◆ Sintaxis de fácil comprensión.

En los últimos años, XML ha destacado como la principal tecnología para el almacenamiento de datos de cualquier tipo de complejidad y para el intercambio de datos entre aplicaciones, o incluso entre diferentes sistemas operativos. Debido al alto grado de interoperabilidad que permite, el XML se utiliza prácticamente en cualquier parte de .NET Framework.

.NET Framework soporta distintas técnicas para trabajar con datos XML. Dispone de clases para la lectura y escritura de archivos XML, clases para la carga de datos XML en memoria, etc., e incluso dispone de clases que implementan el Modelo de Objeto Documento XML (DOM).

### 6.2.2.DOM

El DOM (Document Object Model) es un modelo de objetos estandarizado para documentos HTML y XML que define la estructura lógica de los documentos y el modo en que se accede y manipula un documento. Con DOM se puede modificar el contenido, la estructura y el estilo o presentación de los documentos mediante llamadas a funciones y procedimientos que permiten acceder, cambiar, añadir o borrar nodos de información de los documentos XML.

El DOM es una especificación de W3C, y su objetivo principal es proporcionar una interfaz estándar de programación que pueda utilizarse en una amplia variedad de entornos y aplicaciones independientemente del lenguaje de programación con el que se quiera implementar.

El Dom está basado en la programación orientada a objetos, por lo que no define una estructura de datos, sino que define objetos con los cual crea la estructura jerárquica del documento XML donde se pueden identificar:

- ◆ Las interfaces y los objetos usados para representar y manipular un documento.
- ◆ La semántica de las interfaces y los objetos, incluyendo comportamiento y atributos.
- ◆ Las relaciones y colaboraciones entre las interfaces y los objetos.

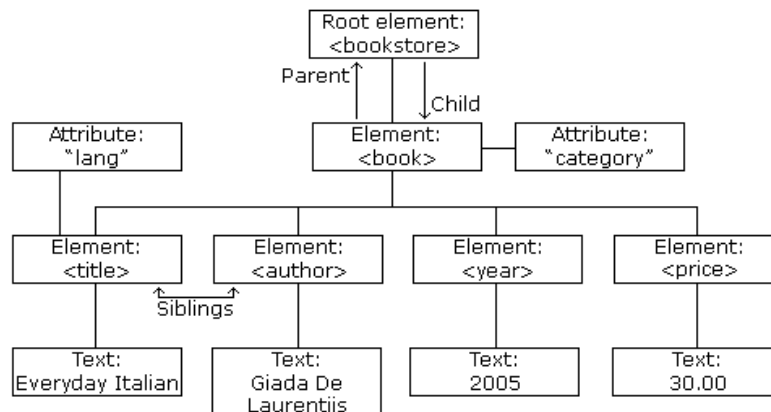


Imagen 53: Ejemplo de la representación de un documento XML propuesta por DOM

El DOM no define la semántica interna real de los lenguajes XML o HTML, sino que es un modelo de programación diseñado para respetar las semánticas establecidas por el W3C en otras especificaciones.

Las especificaciones del W3C que definen DOM son las siguientes:

- ◆ **DOM Nivel 1.** Esta especificación define el Nivel 1 del DOM como una interfaz independiente de la plataforma y del lenguaje que permite a programas y scripts acceder y actualizar dinámicamente los contenidos, la estructura y el estilo de los documentos. La especificación del Nivel 1 del DOM se divide en dos partes: Núcleo y HTML.
  - Especificación del núcleo. La sección sobre el Núcleo del Nivel 1 del DOM proporciona un conjunto de interfaces fundamentales de bajo nivel que pueden representar cualquier documento estructurado, al mismo tiempo que define interfaces extendidas para representar documentos XML. Las implementaciones conformes con el DOM que implementen las interfaces extendidas XML deben también implementar las interfaces del Núcleo, pero no las interfaces HTML. [DOM1]
  - Especificación de HTML. La sección sobre el Nivel 1 de HTML proporciona interfaces adicionales de alto nivel que se utilizan con las interfaces fundamentales definidas en la sección sobre el Núcleo del Nivel 1 para proporcionar una visión más conveniente de los documentos HTML. Una implementación conforme del DOM HTML implementa todas las interfaces fundamentales del Núcleo así como las interfaces HTML. [DOM1]
- ◆ **DOM Nivel 2.** Este nivel se construye sobre el Nivel 1 del DOM. Agrupa una serie de especificaciones, las cuales añaden un conjunto de interfaces especializadas para crear y manipular la estructura y el contenido del documento.
  - Especificación del núcleo. Esta especificación define el núcleo del Nivel 2 del DOM como una interfaz independiente de la plataforma y del lenguaje que permite a

programas y scripts acceder y actualizar dinámicamente los contenidos y la estructura de los documentos. El núcleo del Nivel 2 del DOM está basado en el núcleo del Nivel 1 del DOM. El núcleo del Nivel 2 del DOM está formado por un conjunto de interfaces de núcleo que permiten crear y manipular la estructura y los contenidos de un documento. El núcleo también contiene interfaces especializadas dedicadas a XML. [DOM2n]

- Especificación de las vistas. Define las vistas del Nivel 2 del DOM como una interfaz independiente de la plataforma y del lenguaje que permite a programas y scripts acceder y actualizar dinámicamente el contenido de una representación de los documentos. Las vistas del Nivel 2 del DOM se basan en el núcleo del Nivel 2 del DOM. [DOM2v]
- Especificación de los eventos. Define los eventos del Nivel 2 del DOM como una interfaz independiente de la plataforma y del lenguaje que aporta a programas y scripts un sistema de evento genérico. Los eventos del Nivel 2 del DOM se basan en el núcleo del Nivel 2 del DOM y en las vistas del Nivel 2 del DOM. [DOM2e]
- Especificación del estilo. Esta especificación define las hojas de estilo y las hojas de estilo en cascada (CSS) del Nivel 2 del DOM como una interfaz independiente de la plataforma y del lenguaje que permite a programas y scripts acceder y actualizar dinámicamente el contenido de las hojas de estilo de los documentos. El estilo del Nivel 2 del DOM se basa en el núcleo del Nivel 2 del DOM y en las vistas del Nivel 2 del DOM. [DOM2s]
- Especificación del transversal y el rango. Define el transversal y el rango del Nivel 2 del DOM como una interfaz independiente de la plataforma y del lenguaje que permite a programas y scripts atravesar e identificar dinámicamente un rango de contenido en un documento. El transversal y el rango del Nivel 2 del DOM se basan en el núcleo del Nivel 2 del DOM. La especificación del transversal y el rango del Nivel 2 del DOM está compuesta de dos módulos que contienen las interfaces especializadas dedicadas a atravesar la estructura del documento e identificar y manipular un rango en un documento. [DOM2t]
- Especificación HTML. Define el HTML del Nivel 2 del DOM como una interfaz independiente de la plataforma y del lenguaje que permite a programas y scripts acceder y actualizar dinámicamente el contenido y estructura de documento HTML 4.01 y XHTML 1.0. El HTML del Nivel 2 del DOM se basa en el núcleo del Nivel 2 del DOM y no es compatible con el HTML del Nivel 1 del DOM. [DOM2h]
- ◆ **DOM Nivel 3.** Este nivel se construye sobre el Nivel 2 del DOM para completar el mapeado entre DOM y el conjunto de información XML. [DOM3]
  - Especificación del núcleo. Esta especificación define el núcleo del Nivel 3 del DOM como una interfaz independiente de la plataforma y del lenguaje que permite a programas y scripts acceder y actualizar dinámicamente el contenido, estructura y

estilo de documentos. El núcleo del Nivel 3 del DOM se basa en el núcleo del Nivel 2 del DOM. Esta versión mejora el núcleo del Nivel 2 del DOM para completar el mapeado entre DOM y el conjunto de información XML. [DOM3n]

- Especificación de cargar y guardar. Define el cargar y guardar del Nivel 3 del DOM como una interfaz independiente de la plataforma y del lenguaje que permite a programas y scripts dinámicamente cargar el contenido de un documento XML en un documento DOM y adapte un documento DOM en un documento XML; los documentos DOM son definidos en el núcleo del Nivel 2 del DOM, y los documentos XML son definidos en XML 1.0. [DOM3c]
- Especificación de validación. Define la validación del Nivel 3 del DOM como una interfaz independiente de la plataforma y del lenguaje. Este módulo proporciona la dirección a programas y scripts para actualizar dinámicamente el contenido y la estructura de documentos mientras que se asegura de que el documento sigue siendo válido, o para asegurarse de que el documento llega a ser válido. [DOM3v]

### 6.3.Diseño de la implementación

En este apartado se describe la estructura interna del sistema a implementar.

El sistema utilizará la programación orientada a objetos y la interfaz gráfica se diseñará mediante formularios de Visual Basic .NET. Las consultas y modificaciones a los datos de los documentos XML que harán la función de base de datos se realizarán a través de la interfaz proporcionada por DOM.

La aplicación constará de dos módulos independientes: uno dirigido a las tareas que llevará a cabo el responsable de la evaluación heurística, y otro dirigido a las tareas que realizarán los evaluadores.

Para la implementación de la aplicación se crearán clases propias además de utilizar clases y objetos definidos por la aplicación de desarrollo. Se utilizará la librería System.Xml para trabajar con los documentos XML y la librería System.Net para poder enviar mensajes por correo electrónico. También se utilizará una librería externa iTextSharp para la creación de documentos PDF.

#### 6.3.1.Documentos XML

Como se ha dicho anteriormente, se ha utilizado la tecnología XML para almacenar los datos del gestor en documentos. Se van a tener dos tipos de documentos XML: uno que servirá como base de datos general para el gestor en el módulo del evaluador, y otro que representará la heurística de cada proyecto.

El documento XML que contiene la información del gestor está dividido en tres partes: el apartado de criterios que contiene toda la información de los criterios junto con las subheurísticas correspondientes, el apartado de proyectos que contiene la información de los proyectos y los evaluadores, y el apartado de servidores que contiene los datos de los servidores utilizados para enviar los correos.

El documento XML referente a la heurística también consta de tres apartados: un apartado con la información del proyecto a evaluar, otro apartado con la información de la heurística perteneciente a ese proyecto y los datos de la evaluación y otro con los datos de los servidores.

### 6.3.2. Clases

A continuación se describen brevemente las clases que se han creado para los dos módulos.

#### Clase Proyecto

La clase Proyecto se utilizará para trabajar con los datos de cada proyecto. Sus atributos serán: un identificador del proyecto que se generará automáticamente, el título, el director, una breve descripción y la fecha de registro, además de una lista con todos los proyectos que contiene el gestor. Además el módulo del Evaluador contendrá también en esta clase el nombre del responsable de la evaluación.

Las funciones que pertenecen a esta clase en el módulo del Responsable son de insertar, consultar, modificar y eliminar.

- ◆ Insertar\_proyecto()
- ◆ Consultar\_proyecto()
- ◆ Actualizar\_proyecto()
- ◆ Eliminar\_proyecto()
- ◆ Listar\_proyectos()

Las funciones de esta clase en el módulo del Evaluador son solo de consulta:

- ◆ Consultar\_proyecto()
- ◆ Listar\_proyectos()

#### Clase Evaluador

La clase Evaluador pertenece solo al módulo del responsable y utilizará los datos del evaluador y sus atributos serán: el nombre del evaluador, la dirección de correo electrónico que servirá como identificador, el número de teléfono, una lista con la puntuación de cada criterio.

Las funciones que pertenecen a esta clase son de insertar, consultar, modificar y eliminar.

- ◆ Insertar\_evaluador()
- ◆ Consultar\_evaluador()
- ◆ Actualizar\_evaluador()
- ◆ Eliminar\_evaluador()
- ◆ Listar\_evaluadores()
- ◆ Listar\_evaluadores\_enviar()
- ◆ puntuación\_evaluador()
- ◆ estado\_enviar()
- ◆ estado\_recibir()

### **Clase Criterio**

La clase Criterio trabajará con los datos de los criterios y sus atributos serán: el título del criterio que servirá como identificador, el nombre completo del criterio, la descripción, la observación del evaluador respecto a ese criterio, y una lista de los criterios.

En el módulo del Responsable, las funciones que pertenecen a esta clase son de insertar, consultar, modificar y eliminar.

- ◆ Insertar\_criterio()
- ◆ Eliminar\_criterio()
- ◆ Observacion\_criterio()
- ◆ ConsultarNombre\_criterio()
- ◆ Consultar\_criterio()
- ◆ Listar\_TitCriterio()

Las funciones de esta clase en el módulo del Evaluador son solo de consulta:

- ◆ Consultar\_criterio()
- ◆ ConsultarNombre\_criterio()
- ◆ Listar\_TitCriterio()

### **Clase Pregunta**

La clase Pregunta utilizará los datos de las subheurística y sus atributos serán: el identificador de la pregunta que se genera automáticamente, el texto de la pregunta, el criterio al cual pertenece, una lista de preguntas según el criterio y una lista de todas las preguntas de la heurística. En el módulo del Evaluador también habrá un atributo que haga referencia a la puntuación de la pregunta.

En el módulo del Responsable, las funciones que pertenecen a esta clase son de insertar, consultar, modificar y eliminar.



- ◆ Consultar\_pregunta()
- ◆ Listar\_preguntas()
- ◆ Listar\_preguntas\_criterio()
- ◆ Insertar\_pregunta()
- ◆ Eliminar\_pregunta()
- ◆ Eliminar\_preguntas\_extras()
- ◆ Pregunta\_marcada()

Las funciones de esta clase en el módulo del Evaluador son solo de consulta:

- ◆ Consultar\_pregunta()
- ◆ Listar\_preguntas()

### **Clase Heurística**

La clase Heurística hace referencia a la heurística que servirá para hacer la evaluación. Esta clase tiene los siguientes atributos: uno que hace referencia a si se ha enviado la heurística al evaluador, otro a si se ha recibido la heurística del evaluador y una lista con la puntuación total de cada criterio.

Para el módulo del Responsable las funciones definidas para esta clase son:

- ◆ Crear\_heuristica()
- ◆ Guardar\_heuristica()
- ◆ Eliminar\_heuristica()
- ◆ Descargar\_heuristica()
- ◆ Imprimir\_resultados()
- ◆ Estado\_heuristica()
- ◆ Calcular\_puntos()

Para el módulo del Evaluador las funciones que tiene la clase son:

- ◆ Guardar\_evaluacion()
- ◆ Inicializar()
- ◆ Imprimir\_heuristica()
- ◆ Existe\_evaluacion()
- ◆ Puntos\_sin\_rellenar()
- ◆ Descargar\_evaluacion()

### **Clase Email**

La clase Email hace referencia al envío por correo electrónico de los documentos XML. Esta clase tiene como atributos: la dirección del correo electrónico de la persona que envía el documento,

la contraseña de su correo electrónico, el nombre del servidor, la dirección del servidor de correo de salida, el puerto del servidor, el asunto del correo y el mensaje.

Las funciones de esta clase son las mismas en ambos módulos:

- ◆ Consultar\_servidor()
- ◆ Insertar\_servidor()
- ◆ Copiar\_servidores()
- ◆ Eliminar\_servidor()
- ◆ Enviar\_heuristica()

## 7. Conclusión y futuros trabajos

### Conclusión

Una vez concluido el proyecto, puedo decir que me ha gustado mucho hacer un trabajo de estas características, desarrollar una aplicación de inicio a fin.

El haber utilizado un lenguaje desconocido como Visual Basic .NET y la tecnología XML han hecho que la implementación del sistema fuese más costosa de lo que imaginaba, pero una vez con la lección aprendida todo ha sido más sencillo.

El hecho de realizar pruebas constantes, ha permitido que la aplicación sea más completa de lo que en un principio se esperaba y pueda satisfacer mejor las necesidades de los usuarios.

Me gustaría resaltar la importancia de hacer una buena planificación temporal del desarrollo del proyecto y de tener en cuenta los posibles riesgos que puedan aparecer, puesto que estos imprevistos han hecho que la entrega del proyecto se aplazase más de lo esperado.

Las pruebas de la aplicación con usuarios que se tenían preparadas, no se han podido realizar debido a una restricción de acceso a los archivos en los ordenadores de la universidad que ya se ha solucionado.

En este proyecto no se realiza un análisis exhaustivo de los resultados, ya que no era un objetivo principal en el desarrollo del proyecto. Determinar cual de las diferentes maneras para realizar el análisis sería la más adecuada y la complejidad que lleva mecanizar algo tan subjetivo como las opiniones y posibles soluciones que puedan aportar los evaluadores durante el análisis de los resultados, requiere un estudio detallado de esta actividad. Por eso mismo, el análisis que realiza en estos momentos la aplicación es simplemente una media ponderada de la puntuación de todas las subheurísticas de los criterios.

Estoy convencida de que una aplicación de este tipo, puede tener una buena salida en el mercado y favorecer a los sistemas que la utilicen en su evaluación, ya que la importancia de la usabilidad está cada vez más presente en las empresas dedicadas al desarrollo de software.

Esta aplicación se empleará posteriormente en el bloque y en el máster de IPO y se podrá acceder a él desde la página: [www.mpiua.net](http://www.mpiua.net) -> recursos -> software.

### Futuros trabajos

A pesar de terminar este proyecto, todavía quedan algunos aspectos de la aplicación que se podrían mejorar. Estos aspectos son:

- ♦ Realizar las pruebas de la aplicación con usuarios.
- ♦ Poder gestionar más de una heurística por proyecto.
- ♦ Realizar un análisis completo y eficaz de la evaluación heurística.
- ♦ Compartir los documentos XML de ambos módulos mediante un servidor y no por correo electrónico.

## 10. Bibliografía

- [AccWeb] <http://www.accesibilidadweb.com/?gclid=CMAqqOvOwosCFR2UEAodVRNGAA>
- [Bla05] Luis Miguel Blanco. *Programación en Visual Basic .NET*. Grupo Eidos, 2005.
- [Dix93] A. Dix. *Human computer interaction*, Prentice Hall, 1993
- [DOM1] Vidur Apparao, Steve Byrne y otros. *Document Object Model (DOM) Level 1*. W3C Recommendation 1 October, 1998.  
<http://www.w3.org/TR/1998/REC-DOM-Level-1-19981001/>
- [DOM2n] Arnaud Le Hors, Philippe Le Hégarret y otros. *Document Object Model (DOM) Level 2 Core*. W3C Recommendation 13 November, 2000.  
<http://www.w3.org/TR/2000/REC-DOM-Level-2-Core-20001113/>
- [DOM2v] Arnaud Le Hors, Laurence Cable. *Document Object Model (DOM) Level 2 Views*. W3C Recommendation 13 November, 2000.  
<http://www.w3.org/TR/2000/REC-DOM-Level-2-Views-20001113/>
- [DOM2e] Tom Pixley. *Document Object Model (DOM) Level 2 Events*. W3C Recommendation 13 November, 2000.  
<http://www.w3.org/TR/2000/REC-DOM-Level-2-Events-20001113/>
- [DOM2s] Chris Wilson, Philippe Le Hégarret, Vidur Apparao. *Document Object Model (DOM) Level 2 Style*. W3C Recommendation 13 November, 2000.  
<http://www.w3.org/TR/2000/REC-DOM-Level-2-Style-20001113/>
- [DOM2t] Joe Kesselman, Jonathan Robie y otros. *Document Object Model (DOM) Level 2 Traversal and Range*. W3C Recommendation 13 November, 2000.  
<http://www.w3.org/TR/2000/REC-DOM-Level-2-Traversal-Range-20001113/>
- [DOM2h] Johnny Stenback, Philippe Le Hégarret y otros. *Document Object Model (DOM) Level 2 HTML*. W3C Recommendation 9 January, 2003.  
<http://www.w3.org/TR/2003/REC-DOM-Level-2-HTML-20030109/>
- [DOM3n] Arnaud Le Hors, Philippe Le Hégarret. *Document Object Model (DOM) Level 3 Core*. W3C Recommendation 7 April, 2004.  
<http://www.w3.org/TR/2004/REC-DOM-Level-3-Core-20040407/>
- [DOM3c] Johnny Stenback. *Document Object Model (DOM) Level 3 Load and Save*. W3C Recommendation 7 April, 2004.  
<http://www.w3.org/TR/2004/REC-DOM-Level-3-LS-20040407/>
- [DOM3v] Ben Chang. *Document Object Model (DOM) Level 3 Validation*. W3C Recommendation 15 December, 2003.  
<http://www.w3.org/TR/2004/REC-DOM-Level-3-Val-20040127/>

- [EG01] *Estándares y guías*, 2001
- [GLIPO] Toni Granollers y Jesús Lorés. *Introducción a la IPO*.
- [GLP04] T. Granollers, J. Lorés, A. Perdrix. *Incorporación de Usuarios en la Evaluación de la Usabilidad por Recorrido Cognitivo*. In Proceedings of Interaction'04. 2004
- [GMPIUA] <http://griho.udl.es/mpiuu/iso.htm>
- [ISO98] ISO. *International Standard. ISO 9241-11. Guidance on Usability*, 1998
- [ISO99] ISO. *International Standard. ISO 13407. Human-centred design, 1999 processes for interactive systems.*, 1999.
- [ISO00] ISO. *International Standard. ISO/IEC FDIS 9126-1. Software Engineering - Product quality - Part 1: Quality model* , 2000
- [May99] D.J. Mayhew. *The Usability Engineering Lifecycle: a practitioner's Handbook for User Interface Design*. Academic Press, 1999.
- [MMPG98] M. J. Muller, L. Matheson, C. Page, R. Gallup. *Methods & tools: participatory heuristic evaluation*. In Proceedings of Interaction '98. ACM Press, 1998.
- [MPIUA] T. Granollers y J. Lorés. *El Modelo de Proceso de la Ingeniería de la Usabilidad y de la Accesibilidad*.
- [Nie92] J. Nielsen. *Finding usability problems through heuristic evaluation*. In CHI '92: Proceedings of ACM CHI. 1992
- [Nie93] J. Nielsen. *Usability Engineering*, Academic Press, 1993
- [NL93] J. Nielsen and T. K. Landauer. *A mathematical model of the finding of usability problems*. In CHI '93: Proceedings of ACM CHI. 1993
- [NM90] J. Nielsen and R. Molich. *Heuristic evaluation of user interfaces*. In CHI '90: Proceedings of ACM CHI. 1990
- [NM94] J. Nielsen, R. L. Mack. *Usability Inspection Methods*. John Wiley and Sons, New York, 1994.
- [Press01] R.S. Pressman. *Ingeniería del Software, un enfoque práctico*. McGraw Hill, 2001.
- [Sh86] <http://www.cs.utexas.edu/users/almstrum/cs370/elvisino/rules.html>
- [Tog03] <http://www.asktog.com/basics/firstPrinciples.html>

## Anexo 0. Plantilla de criterios heurísticos

### 0.1.Plantilla utilizada en la aplicación

#### 1. Claridad de los objetivos

La interfaz debe comunicar de manera inmediata su propósito, objetivo y funciones.

- El propósito u objetivo que la interfaz nos transmite es claro y obvio. Con una simple hojeada podemos deducir cuál es su propósito y cuál es su finalidad.
- En el caso de existir más de un objetivo, estos están relacionados con el objetivo o función global y establecen coherencia entre ellos.
- En caso de existir más de un objetivo, estos son claros y separados.

#### 2. Visibilidad del estado del sistema

El sistema debe tener siempre a los usuarios informados del estado del sistema, con una realimentación apropiada y en un tiempo razonable.

- La interfaz (en el caso de tratarse de un sitio web) incluye de forma visible el título del sitio, de la sección o el título de la página.
- Sabes en todo momento dónde estás posicionado.
- Los vínculos están claramente diferenciados.
- No existe información o acciones que necesiten de una acción para su visualización.

#### 3. Adecuación al mundo y a los objetos mentales del usuario / lógica de la información

El sitio/aplicación está adaptado al mundo real de los usuarios, su lenguaje, conocimientos, etc.

- La presentación de los contenidos es familiar o comprensible para el usuario.
- Cuando existen opciones están ordenadas de manera lógica para la forma de pensar del usuario.
- Las metáforas e iconos que utilizan son entendibles para el usuario y facilitan la interacción con la interfaz.
- La interfaz usa el lenguaje del usuario con palabras, frases y conceptos que le son familiares. El lenguaje debe ser claro, simple y con una sola idea por párrafo.
- La información está estructurada con títulos, negritas y viñetas.
- El lenguaje y la disposición de la información es asequible y de lectura rápida para el usuario.
- La estructura y presentación de la información no necesita explicaciones o información adicional para su comprensión.
- Los textos y enunciados de los campos están redactados de forma afirmativa.
- Se utiliza el lenguaje en forma directa, no impersonal.

#### 4. Control y libertad para el usuario

Los usuarios eligen a veces funciones del sistema por error y necesitan a veces una salida de emergencia claramente marcada, eso es salir del estado indeseado sin tener que pasar por un diálogo extendido. Es importante disponer de deshacer y rehacer.

- Se utilizan animaciones no controladas por el usuario.
- El scroll no es más grande que dos pantallas.
- Es posible guardar información de la página Web/sistema interactivo.

- Es posible imprimir la información de la página Web/sistema interactivo sin perder información.
- Existe un vínculo que permite volver a la página inicial.
- Es posible aumentar y disminuir el tamaño de la letra.
- La interfaz se visualiza perfectamente con diferentes resoluciones.
- La interfaz no introduce tecnologías que requieren versiones actualizadas de navegadores o plugs-ins externos.

### **5. Consistencia y estándares**

Los usuarios no han de preguntar si las distintas palabras, situaciones o acciones quieren decir lo mismo. En general siguen las normas y convenciones de la plataforma sobre el que se está implementado el sistema.

- Las etiquetas de los vínculos tienen los mismos nombres que los títulos de las páginas a las que se dirigen.
- Las mismas acciones llevan a los mismos resultados.
- Los mismos elementos son iguales en todo el sitio.
- La misma información (texto) se expresa de la misma forma en toda la página.
- La información está organizada y es mostrada de manera similar en cada página.
- Se utilizan los colores estándares para los vínculos visitado y no visitados.
- Utiliza de manera diferente a la norma, convenciones o etiquetas universales.
- Las áreas de navegación superior, laterales, herramientas de búsqueda y controles (botones, radio buttons, ...) siguen los estándares comunes de mercado.

### **6. Prevenir errores**

Es más importante prevenir la aparición de errores que generar buenos mensajes de error.

- El motor de búsqueda tolera errores tipográficos (mayúsculas), ortográficos (acentos) y acepta palabras similares.

### **7. Reconocimiento más que memoria**

La página Web/sistema interactivo se basa en el reconocimiento más que en el recuerdo, que permite al usuario interactuar con el sitio de manera fácil y productiva.

- EL uso de la interfaz no requiere recordar información de interfaces previas para interactuar con ella. Toda la información necesaria para la interacción se encuentra en la interfaz actual.
- Es fácil localizar información previamente encontrada.
- La interfaz permite ver y seleccionar, más que recordar y escribir.
- La información está organizada según una lógica reconocida y familiar para el usuario.
- Se utilizan iconos relacionados con los contenidos a los que se asocian.
- La estructura, orden y lógica es familiar e intuitiva para los usuarios.

### **8. Flexibilidad y eficiencia de uso**

La interfaz facilita y optimiza el acceso a los usuarios independientemente de cuales sean sus características.

- Existen aceleradores de teclado para realizar operaciones frecuentes.
- Si existen, quedan claros cuales son estos aceleradores.
- Es posible repetir una acción ya realizada anteriormente de manera sencilla.
- Se utiliza un diseño líquido (una página tiene diseño líquido cuando al cambiar de resolución la página se adapta a la pantalla) para que la página se adapte a las diferentes

resoluciones posibles que pueda tener un usuario.

### **9. Diálogos estéticos y diseño minimalista**

La página Web/sistema interactivo evita toda información o gráfico irrelevante y sólo incluye la información necesaria.

- La información visible es la única esencial para realizar la acción. La página no contiene información que es irrelevante o raramente necesaria.
- No existe redundancia de información en la página.
- La información es corta, concisa y precisa.
- Cada elemento de información se distingue del resto y no se confunde con otros.
- El texto es fácil de hojear, está bien organizado y las frases no son muy largas.
- Las fuentes son legibles y tienen un tamaño adecuado.
- Las fuentes utilizan colores con suficiente contraste con el fondo.

### **10. Ayuda y documentación**

Aunque es mejor si el sistema puede utilizarse sin documentación, puede ser necesario disponer de ayuda y documentación. Esta debe ser fácil de buscar, centrada con las tareas del usuario, tener información de las etapas a realizar y que no sean muy extensas.

- En caso de existir ayuda, es visible y fácil de encontrar.
- La documentación de ayuda es sensible al contexto, se refiere a la sección donde se encuentra el usuario.
- La documentación de ayuda sobre accesibilidad esta adaptada a las necesidades del usuario.
- La ayuda esta orientada a los objetivos del usuario (generalmente a la resolución de problemas).
- La página dispone de un apartado de preguntas frecuentes.
- La documentación de ayuda utiliza ejemplos.

### **11. Búsqueda**

Este heurístico se aplica si existe un buscador en la interfaz (normalmente para interfaces web).

- En la página de inicio existe un cuadro de texto para introducir palabras a buscar en el sitio.
- El cuadro de entrada de texto para buscar ocupa entre 25 y 30 caracteres visibles.
- El área de búsqueda esta identificada con una cabecera que titula la opción de búsqueda.

### **12. Noticias**

Este heurístico se aplica para aquellas interfaces que incluyen un apartado específico para las noticias.

- Los titulares de las noticias son breves y descriptivos.
- Existen resúmenes específicos en las noticias (que no incluyen solamente las primeras líneas o el primer párrafo de la noticia).
- El vínculo principal a la noticia ampliada se encuentra en los titulares de las noticias.

### **13. Varios**

Los usuarios deben saber que la información es actual.

- Se muestra la fecha de la última actualización.
- Las fechas se muestran en formato internacional.



## 14. Arquitectura de la información

Organización la información del sitio Web/sistema interactivo.

- El número de pestañas es adecuado.
- Los niveles de profundidad de información\* existentes son suficientes.

## 0.2. Documento Excel utilizado

Datos del evaluador

	A	B	C	D	E	F	G	H
1	<b>Datos del evaluador</b>							
2								
3	Nombre							
4	Perfil*							
5	Estudios							
6								
7	* Indicar el perfil profesional del evaluador (profesor, estudiante, desarrollador de aplicaciones, etc.) y su especialización							
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								
23								

*Imagen 1: Documento Excel - Evaluador*

## Leyenda del documento

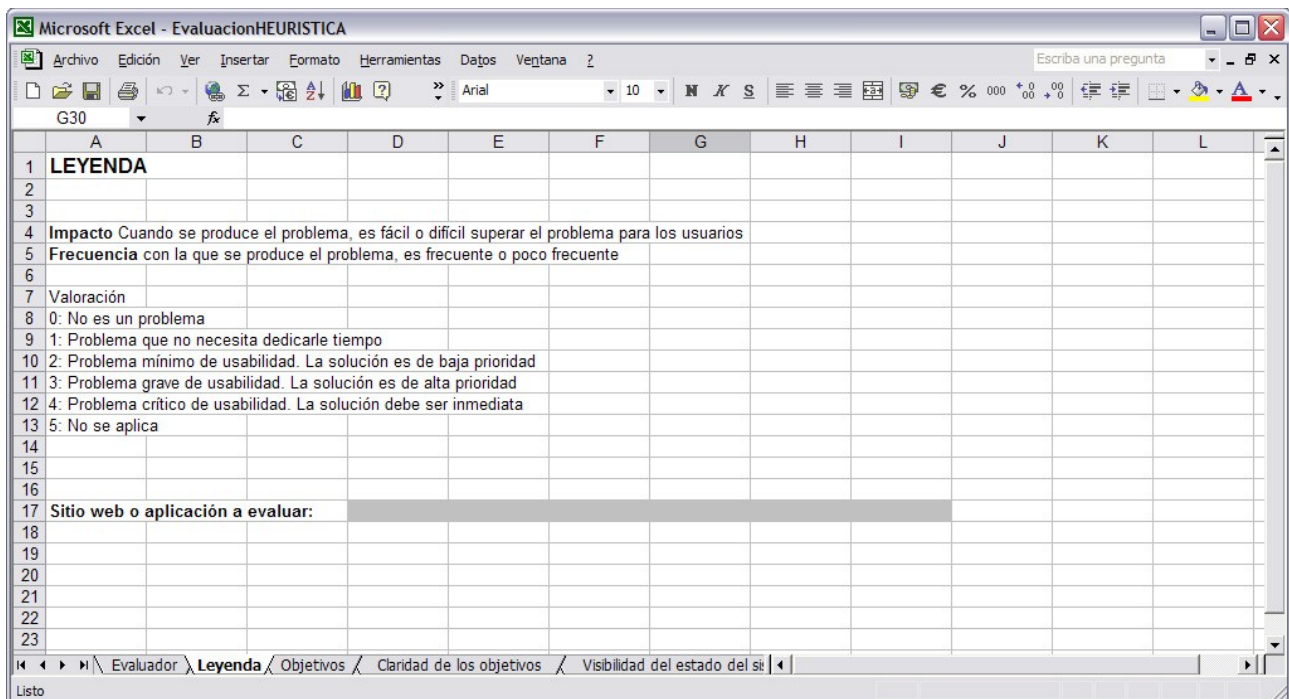


Imagen 2: Documento Excel - Leyenda

## Objetivos de la aplicación

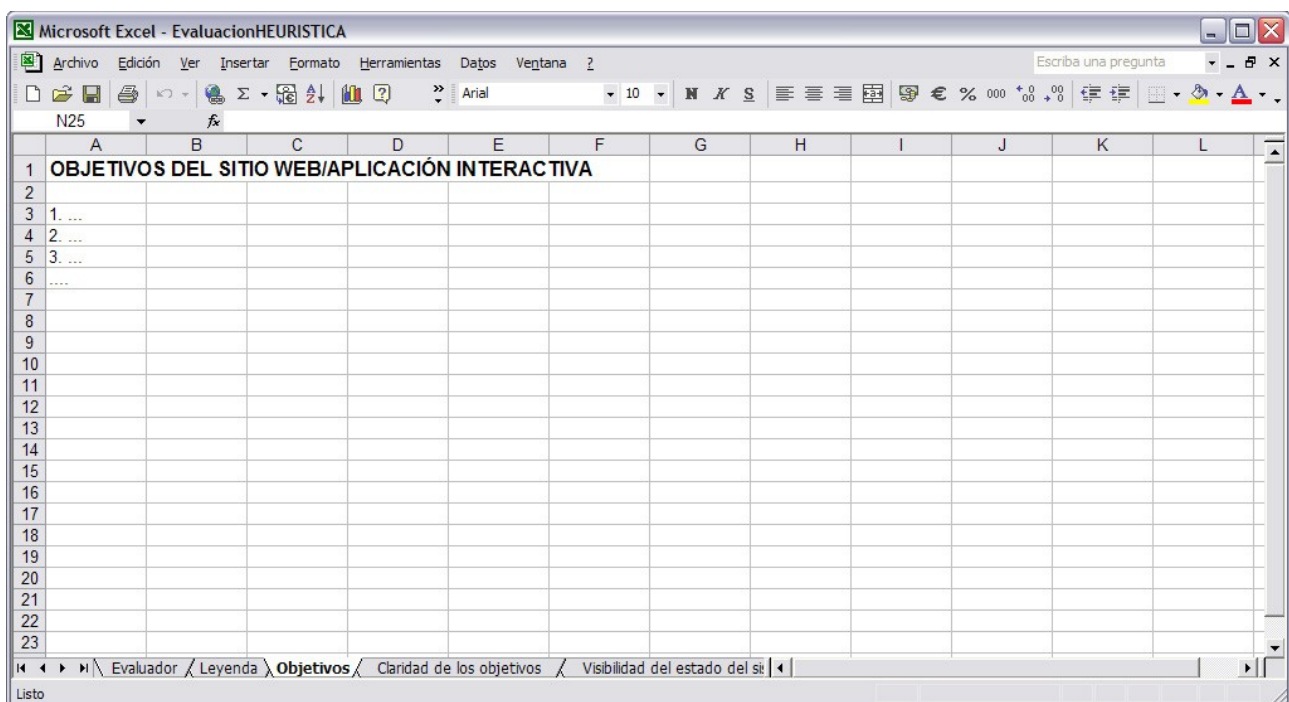
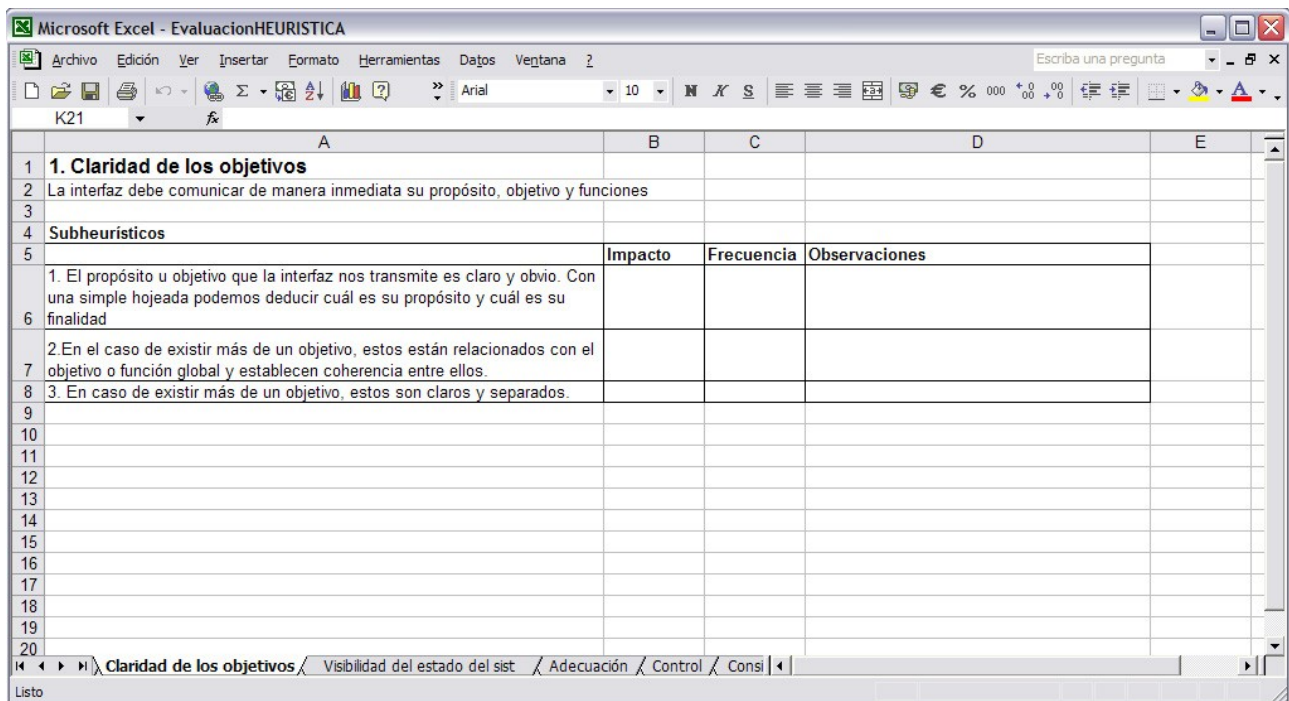


Imagen 3: Documento Excel - Objetivos

## Criterio 1: Claridad de los objetivos



Microsoft Excel - EvaluacionHEURISTICA

Archivo Edición Ver Insertar Formato Herramientas Datos Ventana ?

Escriba una pregunta

K21

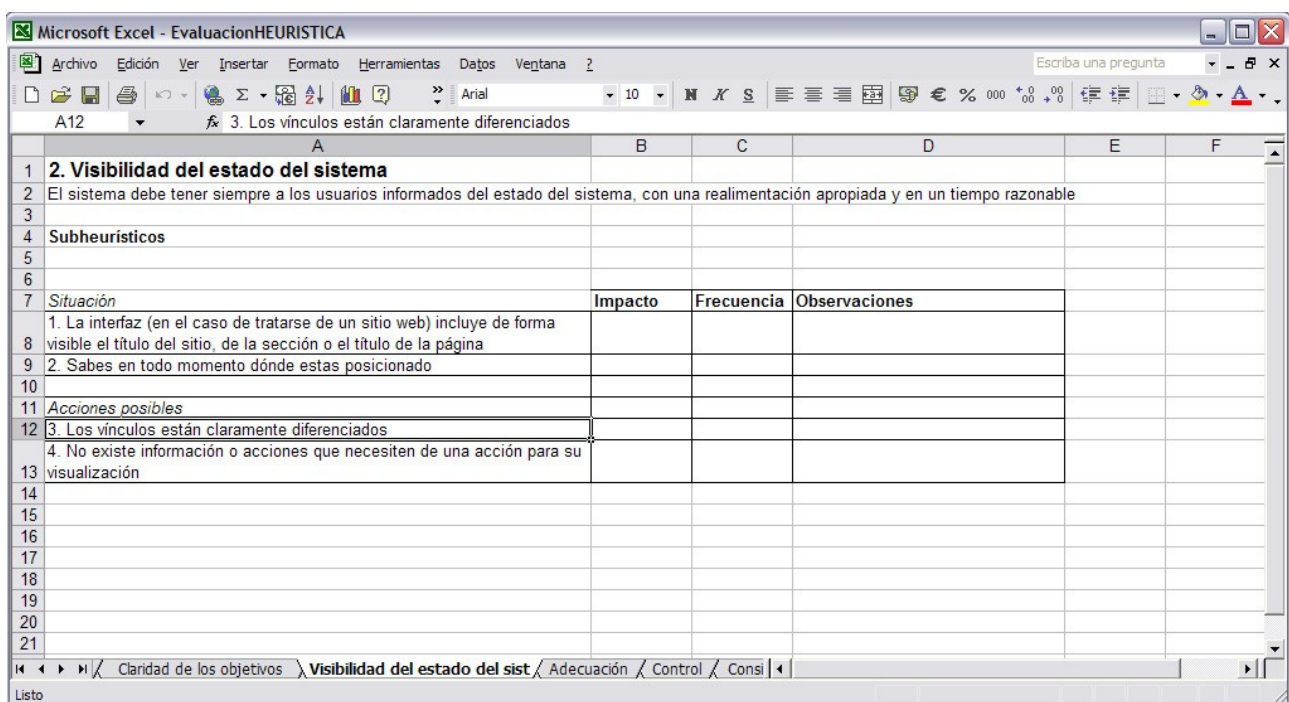
	A	B	C	D	E
1	<b>1. Claridad de los objetivos</b>				
2	La interfaz debe comunicar de manera inmediata su propósito, objetivo y funciones				
3					
4	<b>Subheurísticos</b>				
5		<b>Impacto</b>	<b>Frecuencia</b>	<b>Observaciones</b>	
6	1. El propósito u objetivo que la interfaz nos transmite es claro y obvio. Con una simple hojeda podemos deducir cuál es su propósito y cuál es su finalidad				
7	2. En el caso de existir más de un objetivo, estos están relacionados con el objetivo o función global y establecen coherencia entre ellos.				
8	3. En caso de existir más de un objetivo, estos son claros y separados.				
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					

Claridad de los objetivos Visibilidad del estado del sist Adecuación Control Cons

Listo

Imagen 4: Documento Excel - Claridad

## Criterio 2: Visibilidad del estado del sistema



Microsoft Excel - EvaluacionHEURISTICA

Archivo Edición Ver Insertar Formato Herramientas Datos Ventana ?

Escriba una pregunta

A12

	A	B	C	D	E	F
1	<b>2. Visibilidad del estado del sistema</b>					
2	El sistema debe tener siempre a los usuarios informados del estado del sistema, con una realimentación apropiada y en un tiempo razonable					
3						
4	<b>Subheurísticos</b>					
5						
6						
7	<b>Situación</b>	<b>Impacto</b>	<b>Frecuencia</b>	<b>Observaciones</b>		
8	1. La interfaz (en el caso de tratarse de un sitio web) incluye de forma visible el título del sitio, de la sección o el título de la página					
9	2. Sabes en todo momento dónde estás posicionado					
10						
11	<b>Acciones posibles</b>					
12	3. Los vínculos están claramente diferenciados					
13	4. No existe información o acciones que necesiten de una acción para su visualización					
14						
15						
16						
17						
18						
19						
20						
21						

Claridad de los objetivos Visibilidad del estado del sist Adecuación Control Cons

Listo

Imagen 5: Documento Excel - Visibilidad

## Criterio 3: Adecuación al mundo y a los objetos mentales del usuario / lógica de la información

3. Adecuación al mundo y a los objetos mentales del usuario / lógica de la información				
El sitio/aplicación esta adaptado al mundo real de los usuarios, su lenguaje, conocimientos, etc.				
Subheurísticos				
Lógica de la información	Impacto	Frecuencia	Observaciones	
1. La presentación de los contenidos es familiar o comprensible para el usuario				
2. Cuando existen opciones están ordenadas de manera lógica para la forma de pensar del usuario				
3. Las metáforas e iconos que utilizan son entendibles para el usuario y facilitan la interacción con la interfaz.				
Formato de la información				
4. La interfaz usa el lenguaje del usuario con palabras, frases y conceptos que le son familiares. El lenguaje debe ser claro, simple y con una sola idea por párrafo				
5. La información esta estructurada con títulos, negritas y viñetas				
6. El lenguaje y la disposición de la información es asequible y de lectura rápida para el usuario				
7. La estructura y presentación de la información no necesita explicaciones o información adicional para su comprensión				
8. Los textos y enunciados de los campos están redactados de forma afirmativa				
9. Se utiliza el lenguaje en forma directa, no impersonal				

Imagen 6: Documento Excel - Adecuación

## Criterio 4: Control y libertad para el usuario

4. Control y libertad para el usuario					
Los usuarios eligen a veces funciones del sistema por error y necesitan a veces una salida de emergencia claramente marcada, eso es salir del estado indeseado sin tener que pasar por un diálogo extendido. Es importante disponer de deshacer y rehacer					
Subheurísticos					
	Impacto	Frecuencia	Observaciones		
1. Se utilizan animaciones no controladas por el usuario					
2. El scroll no es más grande que dos pantallas					
3. Es posible guardar información de la página Web/sistema interactivo					
4. Es posible imprimir la información de la página Web/sistema interactivo sin perder información					
5. Existe un vínculo que permite volver a la página inicial					
6. Es posible aumentar y disminuir el tamaño de la letra					
7. La interfaz se visualiza perfectamente con diferentes resoluciones					
8. La interfaz no introduce tecnologías que requieren versiones actualizadas de navegadores o plug-ins externos.					

Imagen 7: Documento Excel - Control



## Criterio 5: Consistencia y estándares

7. Utiliza de manera diferente a la norma, convenciones o etiquetas universales				
A	B	C	D	E
<b>5. Consistencia y estándares</b>				
Los usuarios no han de preguntar si las distintas palabras, situaciones o acciones quieren decir lo mismo. En general siguen las normas y convenciones de la plataforma sobre el que se esta implementado el sistema.				
<b>Subheurísticos</b>				
<b>Consistencia</b>	<b>Impacto</b>	<b>Frecuencia</b>	<b>Observaciones</b>	
1. Las etiquetas de los vínculos tienen los mismos nombres que los títulos de las páginas a las que se dirigen				
2. Las mismas acciones llevan a los mismos resultados				
3. Los mismos elementos son iguales en todo el sitio				
4. La misma información (texto) se expresa de la misma forma en toda la página				
5. La información esta organizada y es mostrada de manera similar en cada página				
<b>Estándares</b>				
6. Se utilizan los colores estándares para los vínculos visitado y no visitados.				
<b>7. Utiliza de manera diferente a la norma, convenciones o etiquetas universales</b>				
8. Las áreas de navegación superior, laterales, herramientas de búsqueda y controles (botones, radio buttons, ...) siguen los estándares comunes de mercado.				

Imagen 8: Documento Excel - Consistencia

## Criterio 6: Prevenir errores

6. Prevenir errores					
A	B	C	D	E	F
Es más importante prevenir la aparición de errores que generar buenos mensajes de error					
<b>Subheurísticos</b>					
<b>Impacto</b>	<b>Frecuencia</b>	<b>Observaciones</b>			
1. El motor de búsqueda tolera errores tipográficos (mayúsculas), ortográficos (acentos) y acepta palabras similares					

Imagen 9: Documento Excel - Errores

Microsoft Excel - Evaluacion HEURISTICA

Archivo Edición Ver Insertar Formato Herramientas Datos Ventana ?

Escriba una pregunta

M20

	A	B	C	D	E	F
1	<b>7. Reconocimiento más que memoria</b>					
2	La página Web/sistema interactivo se basa en el reconocimiento más que en el recuerdo, que permite al usuario interactuar con el sitio de manera fácil y productiva					
3						
4	<b>Subheurísticos</b>					
5						
6		<b>Impacto</b>	<b>Frecuencia</b>	<b>Observaciones</b>		
7	1. EL uso de la interfaz no requiere recordar información de interfaces previas para interactuar con ella. Toda la información necesaria para la interacción se encuentra en la interfaz actual.					
8	2. Es fácil localizar información previamente encontrada.					
9	3. La interfaz permite ver y seleccionar, más que recordar y escribir					
10	4. La información esta organizada según una lógica reconocida y familiar para el usuario					
11	5. Se utilizan iconos relacionados con los contenidos a los que se asocian					
12	6. La estructura, orden y lógica es familiar e intuitiva para los usuarios.					
13						
14						
15						
16						
17						
18						
19						
20						

Adecuación / Control / Consistencia / Errores / **Reconocimiento** / Flexibilidad / dialogos

Listo

Microsoft Excel - EvaluacionHEURISTICA

Archivo Edición Ver Insertar Formato Herramientas Datos Ventana ?

Escriba una pregunta

L21

	A	B	C	D	E	F
1	<b>8. Flexibilidad y eficiencia de uso</b>					
2	La interfaz facilita y optimiza el acceso a los usuarios independientemente de cuales sean sus características					
3						
4	<b>Subheurísticos</b>					
5		<b>Impacto</b>	<b>Frecuencia</b>	<b>Observaciones</b>		
6	1. Existen aceleradores de teclado para realizar operaciones frecuentes					
7	2. Si existen, quedan claros cuales son estos aceleradores					
8	3. Es posible repetir una acción ya realizada anteriormente de manera sencilla					
9	4. Se utiliza un diseño líquido* para que la página se adapte a las diferentes resoluciones posibles que pueda tener un usuario.					
10						
11						
12	*Una página tiene diseño líquido cuando al cambiar de resolución la página se adapta a la pantalla					
13						
14						
15						
16						
17						
18						
19						
20						
21						

Adecuación / Control / Consistencia / Errores / Reconocimiento / **Flexibilidad** / dialogos

Listo

78

## Criterio 9: Diálogos estéticos y diseño minimalista

	A	B	C	D	E	F
1	<b>9. Diálogos estéticos y diseño minimalista</b>					
2	La página Web/sistema interactivo evita toda información o gráfico irrelevante y sólo incluye la información necesaria					
3						
4	<b>Subheurísticos</b>					
5		<b>Impacto</b>	<b>Frecuencia</b>	<b>Observaciones</b>		
6	1. La información visible es la única esencial para realizar la acción. La página no contiene información que es irrelevante o raramente necesaria					
7	2. No existe redundancia de información en la página					
8	3. La información es corta, concisa y precisa.					
9	4. Cada elemento de información se distingue del resto y no se confunde con otros					
10	5. El texto es fácil de hojear, esta bien organizado y las frases no son muy largas					
11	6. Las fuentes son legibles y tienen un tamaño adecuado					
12	7. Las fuentes utilizan colores con suficiente contraste con el fondo					
13						
14						
15						
16						
17						
18						
19						
20						
21						
22						

Imagen 12: Documento Excel - Diálogos

## Criterio 10: Ayuda y documentación

	A	B	C	D	E	F
1	<b>10. Ayuda y documentación</b>					
2	Aunque es mejor si el sistema puede utilizarse sin documentación, puede ser necesario disponer de ayuda y documentación. Esta debe ser fácil de buscar, centrada con las tareas del usuario, tener información de las etapas a realizar y que no sean muy extensas.					
3						
4	<b>Subheurísticos</b>					
5		<b>Impacto</b>	<b>Frecuencia</b>	<b>Observaciones</b>		
6	1. En caso de existir ayuda, es visible y fácil de encontrar					
7	2. La documentación de ayuda es sensible al contexto, se refiere a la sección donde se encuentra el usuario					
8	3. La documentación de ayuda sobre accesibilidad esta adaptada a las necesidades del usuario					
9	4. La ayuda esta orientada a los objetivos del usuario (generalmente a la resolución de problemas)					
10	5. La página dispone de un apartado de preguntas frecuentes					
11	6. La documentación de ayuda utiliza ejemplos					
12						
13						
14						
15						
16						
17						
18						
19						

Imagen 13: Documento Excel - Ayuda



## Criterio 11: Búsqueda

	A	B	C	D	E
1	<b>11. Búsqueda</b>				
2	Este heurístico se aplica si existe un buscador en la interfaz (normalmente para interfaces web)				
3	<b>Subheurísticos</b>				
4		<b>Frecuencia</b>	<b>Impacto</b>	<b>Observaciones</b>	
5	1. En la página de inicio existe un cuadro de texto para introducir palabras a buscar en el sitio				
6	2. El cuadro de entrada de texto para buscar ocupa entre 25 y 30 caracteres visibles				
7	3. El área de búsqueda esta identificada con una cabecera que titula la opción de búsqueda				
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					

Imagen 14: Documento Excel - Búsqueda

## Criterio 12: Noticias

	A	B	C	D	E	F
1	<b>12. Noticias</b>					
2	Este heurístico se aplica para aquellas interfaces que incluyen un apartado específico para las noticias.					
3	<b>Subheurísticos</b>					
4		<b>Impacto</b>	<b>Frecuencia</b>	<b>Observaciones</b>		
5	1. Los titulares de las noticias son breves y descriptivos					
6	2. Existen resúmenes específicos en las noticias (que no incluyen solamente las primeras líneas o el primer párrafo de la noticia)					
7	3. El vínculo principal a la noticia ampliada se encuentra en los titulares de las noticias					
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						

Imagen 15: Documento Excel - Noticias



## Criterio 13: Varios

	A	B	C	D	E	F
1	<b>13. Varios</b>					
2						
3	<b>Subheurísticos</b>					
4		<b>Frecuencia</b>	<b>Impacto</b>	<b>Observaciones</b>		
5	1. Se muestra la fecha de la última actualización					
6	2. Las fechas se muestran en formato internacional					
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						
22						
23						

Imagen 16: Documento Excel - Varios

## Criterio 14: Arquitectura de la información

	A	B	C	D	E	F	G
1	<b>14. Arquitectura de la información</b>						
2	Organización la información del sitio Web/sistema interactivo						
3							
4	<b>Subheurísticos</b>						
5				<b>Frecuencia</b>	<b>Impacto</b>	<b>Observaciones</b>	
6	1. El número de pestañas es adecuado						
7	2. Los niveles de profundidad de información* existentes son suficientes						
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							



Imagen 17: Documento Excel - Arquitectura

## Anexo 1. Evaluación del prototipo

Resultados de la evaluación de la maqueta digital realizada por los alumnos del Bloque de IPO de Ingeniería Técnica de Gestión de la UdL.

### Alumno1

#### **Aplicación:**

- En la página principal no queda demasiado claro la organización de los botones: *Lista de proyectos* y *Nuevo proyecto* y la información a la que están ligados.
- En *Datos del proyecto* resaltar más los títulos: *Director*, *Descripción proyecto* y *Fecha*.
- Remarcar que el apartado *Estado del proyecto* se modifica en pico se va avanzando en los apartados. Puede ser estaría bien añadir una fecha de finalización de cada parte.
- En los detalles puede ser que no quede claro lo que el  y la  significan: “recibido” y “no recibido”. Puede ser no necesario poner el e-mail con el nombre, si lo queremos averiguar iremos a *Evaluadores*.
- En el apartado *Heurística*, no se ve claro que es la lista expuesta. Puede que se tendría que añadir “Lista de criterios”. Así, a la vez entenderemos el botón que hay dentro del apartado *Empezar Heurística*.
- Cuando clicamos el botón *Enviar heurística*, no se notifica a quien envía la evaluación y si está enviada con éxito.
- Cuando eliminas un proyecto, puede ser que tendría que aparecer una ventana preguntando si estamos completamente seguros.
- No veo diferencia entre modificar y empezar heurística.

#### **Evaluadores:**

- No está demasiado claro los apartados dentro de la información del proyecto.
- Igual que antes necesitaría indicar de que se trata la lista de los apartados de la heurística.

### Alumno2

#### **Aplicación:**

- En la primera página no queda claro que estamos en *Lista de proyectos*.
- En la primera página no es evidente que se pueda clicar sobre *Rediseño Web AIPO* (lo sería en un entorno web pero es poco coherente con el sistema de botones).
- Los botones de *Nuevo proyecto* y *Lista de proyectos* que siempre están visibles crean un cierto desorden. Por ejemplo, en la pestaña *Información* tenemos botones en las cuatro esquinas.
- Los estados del proyecto denotan acciones ha hacer en lugar de acciones hechas. Por ejemplo, *Recibir heurística* en lugar de “Heurística recibida”.
- En la pestaña *Evaluadores* vuelve a aparecer la incoherencia de que algunas acciones se realizan pulsando botones y otras pulsando hipervínculos.
- En la pestaña *Heurística* se da ha entender que se pueda realizar la heurística cuando solo se diseña.

- Al empezar a diseñar la heurística, los botones ► y ►| hacen aparentemente la misma acción.
- En el diseño de la heurística es difícil distinguir en qué apartado nos encontramos *Claridad, Visibilidad...*, pues el contraste es casi nulo.

**Evaluadores:**

- No se sabe donde hay que pulsar para obtener la versión imprimible.
- Igual que en la aplicación, la barra de navegación es poco clara, pues tiene poco contraste.
- Se podría pensar que la palabra *Observaciones* es un hipervínculo (ya que a veces en el programa las hay).
- Los botones ► y ►| hacen lo mismo.

**Alumno3****Aplicación:****INICIO**

- Dos idiomas en la pantalla de inicio, botones castellano, botones catalán.
- Botón de *Atrás* innecesario, ya que puedes acceder a todos los lugares en teoría desde toda la pantalla, tanto sea *Nuevo proyecto*, como *Lista de proyectos*.

**HEURÍSTICA**

- El menú de abajo se podría modificar de manera que se viera más accesible y de manera completa sin tener que mover con la barra o flechas.
- Muy poco espacio para ver la información, el menú de arriba y el de abajo ocupa más de la mitad de la página. Se podría optimizar de manera que ocupase menos y tener más espacio para la información esencial.

**Evaluadores:**

- Misma opinión que en la aplicación sobre el cuerpo del diseño, menús de arriba y abajo bastante grandes.
- Te lleva a una pantalla donde no tienes opción a elegir, ni indicación de si se ha enviado correctamente.

**Alumno4****Aplicación:**



- Confunde un poco el que la página de inicio tenga un formato diferente a la del resto. Quizás sería más intuitivo mantener una estructura básica común a todas las pantallas. Por ejemplo, mostrar siempre las pestañas disponibles, quizás deshabilitando las que todavía no puedan ser utilizadas.
- Al elegir un proyecto, sería interesante poder tener siempre a la vista el proyecto en el que estamos trabajando, para no tener que acordarnos.
- No queda claro como se modifica la heurística, que opciones de edición existen.
- Algunas etiquetas no tienen un mensaje suficientemente claro. Por ejemplo, *Empezar heurística*.

- ¿No se puede tener más de una heurística?
- Cuando registras un nuevo proyecto, te lleva directamente a las pestañas para gestionarlo lo que resulta algo confuso, ya que estábamos en la pantalla de selección de proyecto.
- Se echa en falta un botón de “Inicio” o similar.

**Evaluadores:**

- Si durante la realización de la heurística necesitamos consultar los criterios, mediante el botón *Lista de criterios*, cambia radicalmente el aspecto, resulta algo desconcertante. Además, no tenemos la opción de continuar con la heurística, sino que solo disponemos de la opción *Empezar heurística*. Parece que se perderá el trabajo hecho hasta el momento.
- Por experiencia propia, resulta ser muy necesario los criterios con frecuencia, habría que agilizar tal consulta, o incluso mantenerlo siempre visible.
- ¿Cómo recibimos las heurísticas que nos envía el responsable? ¿Y si estamos trabajando en varios proyectos? ¿No podemos almacenar más de una a la vez?
- En casi todas las pantallas aparece el mensaje de *Versión Imprimible*, pero en ningún momento se nos da la opción de imprimir.
- El botón de *Atrás* que hay en la pestaña *Información proyecto* no tiene demasiada utilidad, en vista de que no se nos permite navegar en esta pestaña. También aplicable al de la pestaña *Instrucciones heurística*.

**Alumno5****Aplicación:**

- Donde se indica el estado del proyecto con  o  quizás sería mejor utilizar tres colores, similar a un semáforo.  
Rojo: Sin realizar.  
Amarillo: Se ha empezado, pero no se ha acabado.  
Verde: Se ha acabado.
- En la pestaña de *Heurística* quizás falta un título, como “Índice”.
- Heurística: el botón *Empezar heurística* no indica que se vaya a seleccionar las subheurísticas. Indicar la fecha de envío, de recibo... Estirar línea de *Datos del proyecto*, así se verá mejor que el botón *Modificar proyecto* pertenece a los datos.

**Evaluadores:**

- Falta el botón *Cancelar* en *Correo*.

**Alumno6****Aplicación:**

- El botón de *Empezar proyecto* lleva a confusión.
- Añadir un campo para indicar si el proyecto está cerrado o poner la duración del proyecto.
- Muchos botones en la parte de abajo de la Heurística. No se sabe si se pueden clicar o solo son títulos.

- Indica si un “Recibir E-Mail” por parte de los evaluadores está a medias o no. Detalles: No aparece nada pondría color amarillo para indicar que aún no se ha acabado por parte de algún evaluador.  
Verde: OK  
Amarillo: A medias  
Rojo: Falta por hacer todo

**Evaluadores:**

- La opción de *Correo* no puede volver hacia atrás.
- Falta en *Correo* botón “Cancelar”.

**Alumno7****Aplicación:**

- El objetivo de cada aplicación no está claro, creemos que tendría que haber una explicación.
- El botón de *Empezar heurística* en el apartado *Heurística* nos confunde ya que no empieza la heurística sino que empieza a escoger los criterios que quiere que se analicen en la heurística personalizada.
- Creemos que la pestaña de *Correo* tendría que estar en otro nivel ya que no merece la misma importancia que las demás.

**Evaluadores:**

- Está bien estructurado y es fácil de entender el procedimiento a seguir.
- Cuando se realiza la heurística se tendría que indicar de alguna forma los aspectos ya analizados, por ejemplo, *Claridad*, *Visibilidad*. Con el fin de que si quieres analizar de forma salteada los aspectos sepas cuales están ya analizados.
- Añadir la posibilidad de guardarlo en formato .pdf.
- Faltaría que se pudiesen guardar todas las heurísticas y tener una lista de proyectos y marcarlos como finalizados, pendientes o a medias.

**Alumno8****Aplicación:**

- Se podría introducir un sistema de búsqueda con tal de localizar fácilmente una heurística en concreto que no se conoce del cierto en que grupo se encuentra.
- Se podría marcar los cambios que se realizan con otro color, o buscar un método para que se visualizasen todos los cambios que se han hecho.
- El título que nos muestra en todo momento en que parte del prototipo nos encontramos, no se entiende demasiado. Las palabras del título no corresponden con el lugar exacto donde estamos y puede causar ambigüedades.
- En la *Lista de criterios* se podría linkar con cada lista de subheurísticas con tal de acceder a una directamente.

- Al guardar la heurística sería interesante que se pudiese observar como una vista previa del que se ha escogido.
- El botón *Empezar heurística* parece que puedas realizar directamente una evaluación heurística.
- Podría salir una breve explicación del que quiere decir el link donde estamos sobrepuestos.

**Evaluadores:**

- Tendría que haber una observación para cada punto de la subheurística, ya que podría haber observaciones para cada punto y entonces estos serían confusos porque podría ser que no las relacionasen con el punto correcto.
- Donde aparecen los criterios o heurísticas principales, debería de estar linkado a cada heurística diferente.
- Cuando ya se está haciendo la evaluación, un botón de “Siguiente heurística” y “Anterior heurística” no estaría mal, porque pueden surgir dudas de como se accede a la siguiente página.
- A parte de tener en cuenta el peso de cada subheurística, también se podría considerar la frecuencia en que se encuentra.

## Anexo 2. Manual de usuario

### Índice

<b>Módulo del responsable.....</b>	<b>78</b>
Pantalla inicial.....	78
Registrar proyecto.....	80
Pestaña Información.....	81
Pestaña Evaluadores.....	82
Registrar evaluador.....	85
Pestaña Heurística.....	86
Selección de subheurísticas.....	87
Enviar heurística.....	88
Leer heurística.....	90
Pestaña Análisis.....	92
<b>Módulo del evaluador.....</b>	<b>94</b>
Pantalla inicial.....	94
Pestaña Información.....	95
Pestaña Instrucciones.....	96
Pestaña Evaluación heurística.....	97
Realizar la evaluación.....	98
Enviar evaluación.....	98

## Índice de pantallas

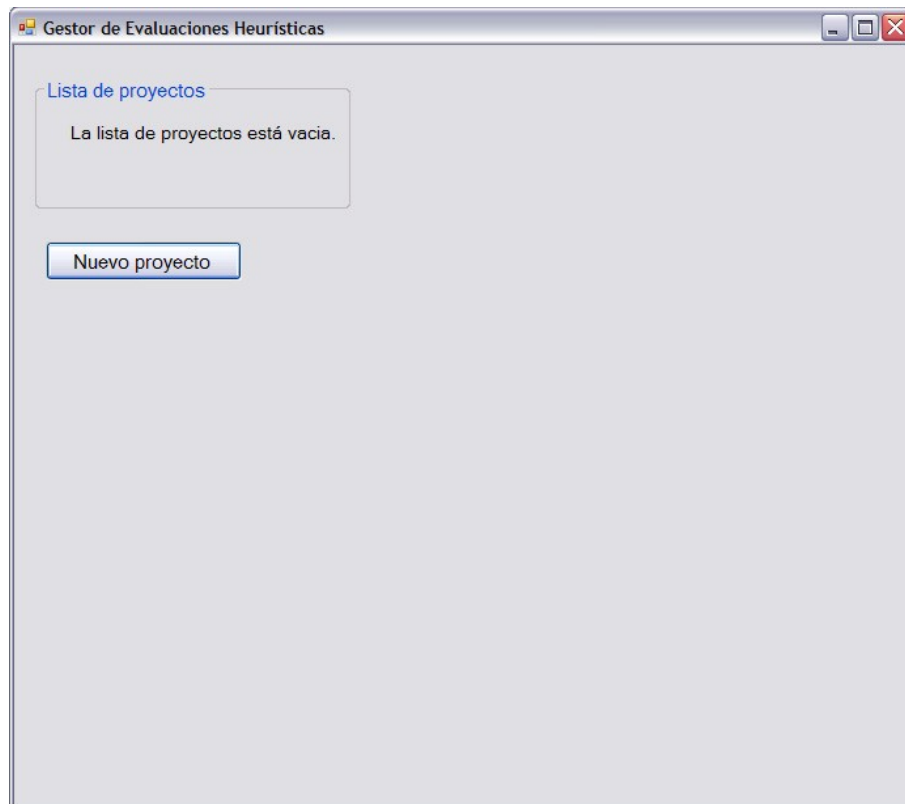
Pantalla 1: Lista de proyectos vacía.....	78
Pantalla 2: Lista de proyectos.....	79
Pantalla 3: Registro del proyecto.....	80
Pantalla 4: Información del proyecto.....	81
Pantalla 5: Pestaña evaluadores vacía.....	82
Pantalla 6: Pestaña evaluadores.....	83
Pantalla 7: Información de un evaluador.....	84
Pantalla 8: Registrar evaluador.....	85
Pantalla 9: Pestaña de la heurística.....	86
Pantalla 10: Criterio de la heurística.....	87
Pantalla 11: Nueva subheurística.....	87
Pantalla 12: Enviar correo.....	88
Pantalla 13: Configuración del servidor de correo.....	89
Pantalla 14: Error al enviar un correo.....	89
Pantalla 15: Pestaña Heurística.....	90
Pantalla 16: Leer heurística.....	91
Pantalla 17: Análisis de la evaluación de un evaluador.....	92
Pantalla 18: Gráfico de usabilidad.....	93
Pantalla 19: Lista de proyectos.....	94
Pantalla 20: Información del proyecto.....	95
Pantalla 21: Instrucciones de la evaluación.....	96
Pantalla 22: Pestaña Evaluación.....	97
Pantalla 23: Evaluar criterio.....	98



## Módulo del Responsable

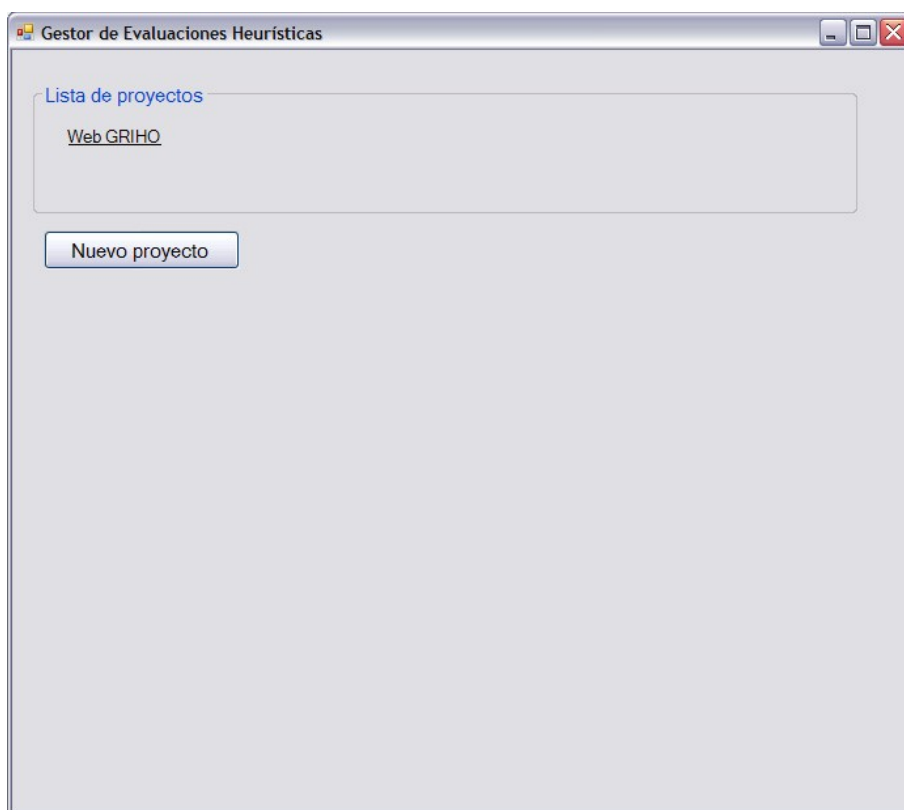
### Pantalla inicial

Cuando ejecutamos el módulo del Responsable de la aplicación por primera vez, aparece una pantalla vacía con un botón para registrar nuevos proyectos.



*Pantalla 1: Lista de proyectos vacía*

Si el gestor de evaluaciones ya contiene proyectos aparecerá una lista con los nombres de los proyectos.

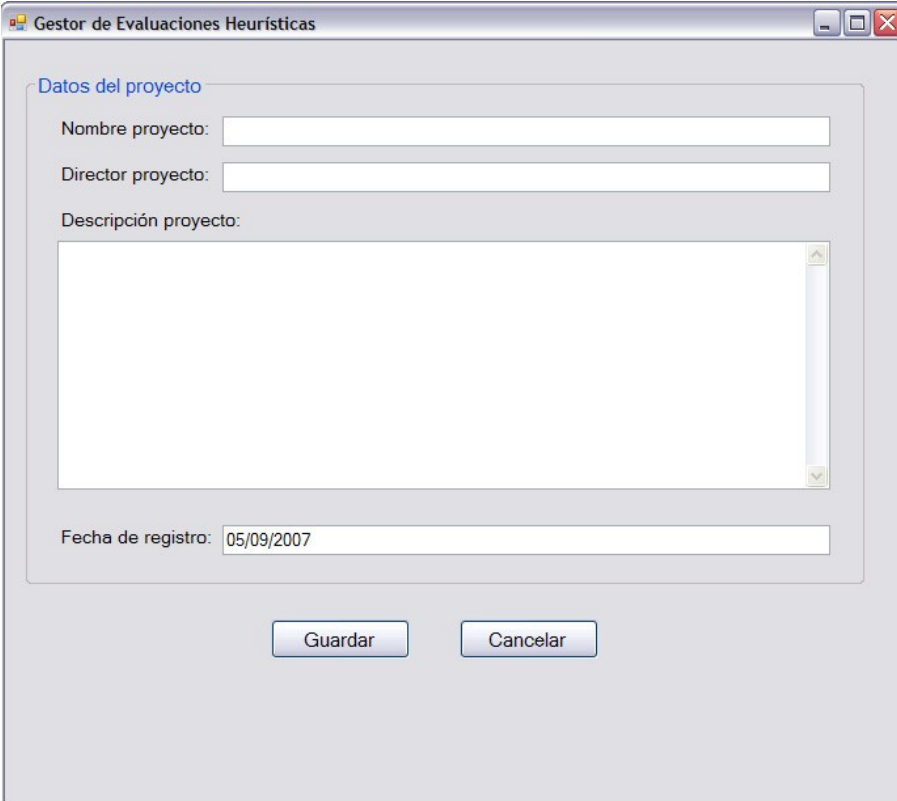


*Pantalla 2: Lista de proyectos*

Para acceder a cada al contenido de un proyecto es necesario presionar en el nombre de éste.

## Registrar proyecto

Una vez presionado el botón de *Nuevo proyecto* en la pantalla inicial, aparece un formulario para introducir los datos del proyecto que se quiere registrar. Para registrar el proyecto es necesario introducir obligatoriamente el nombre de éste. El campo *Fecha de registro* muestra por defecto la fecha actual.



The screenshot shows a window titled "Gestor de Evaluaciones Heurísticas". Inside, there is a section titled "Datos del proyecto" with the following fields:

- Nombre proyecto: [text input field]
- Director proyecto: [text input field]
- Descripción proyecto: [text area]
- Fecha de registro: [text input field containing "05/09/2007"]

At the bottom of the form are two buttons: "Guardar" and "Cancelar".

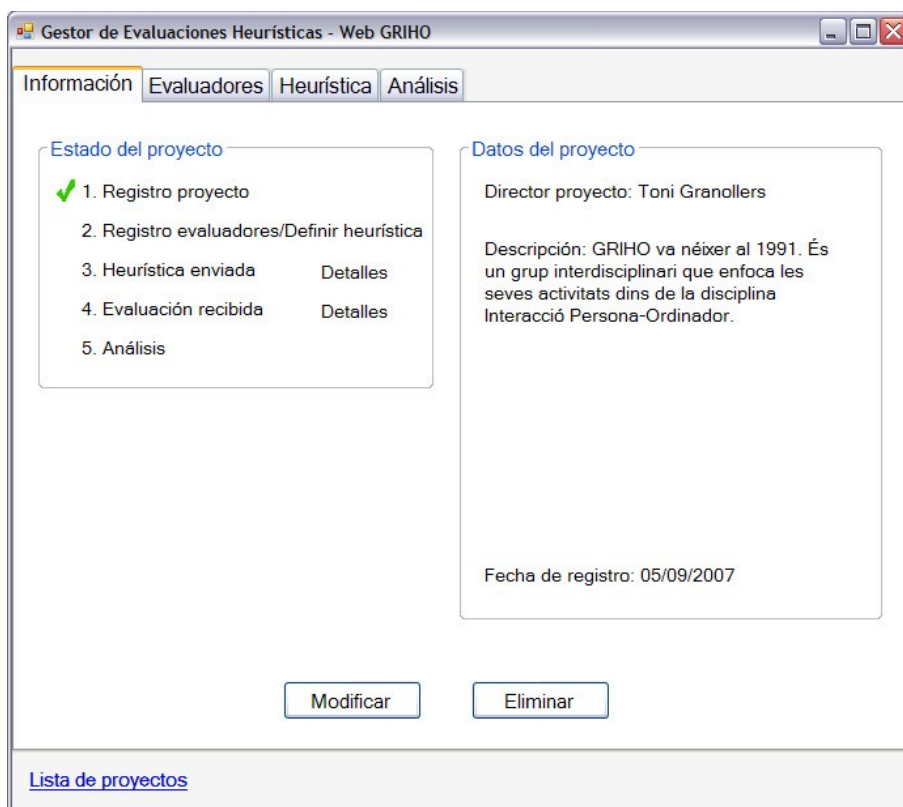
*Pantalla 3: Registro del proyecto*

Este formulario es idéntico al formulario que aparece cuando se quieren modificar los datos del proyecto, pero mostrando por defecto la información guardada anteriormente.

## Pestaña Información

Al acceder a un proyecto, se muestran una serie de pestañas. Cada pestaña representa una actividad de la evaluación perteneciente a la tarea del responsable. Al entrar al proyecto el gestor se sitúa en la pestaña de información del proyecto.

En esta pestaña se muestran los pasos a seguir para realizar la evaluación y en que estado está actualmente. Al lado se muestra la información del proyecto.

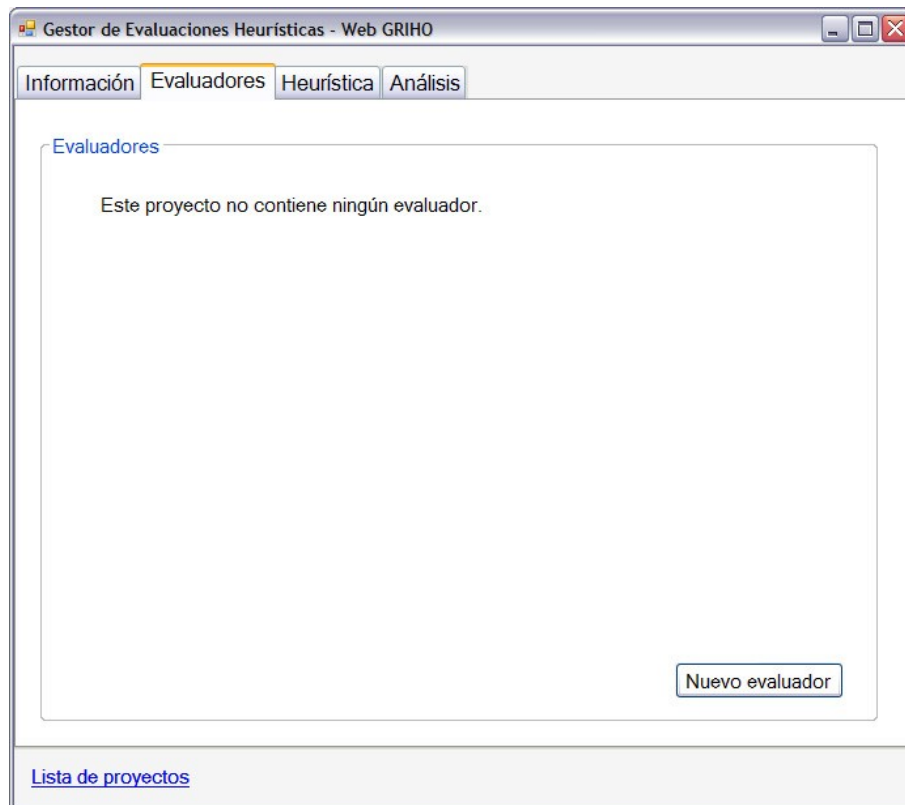


*Pantalla 4: Información del proyecto*

Desde esta pantalla se puede modificar los datos del proyecto o eliminarlo. Al eliminar el proyecto se eliminan también la heurística y las evaluaciones relacionadas con este proyecto.

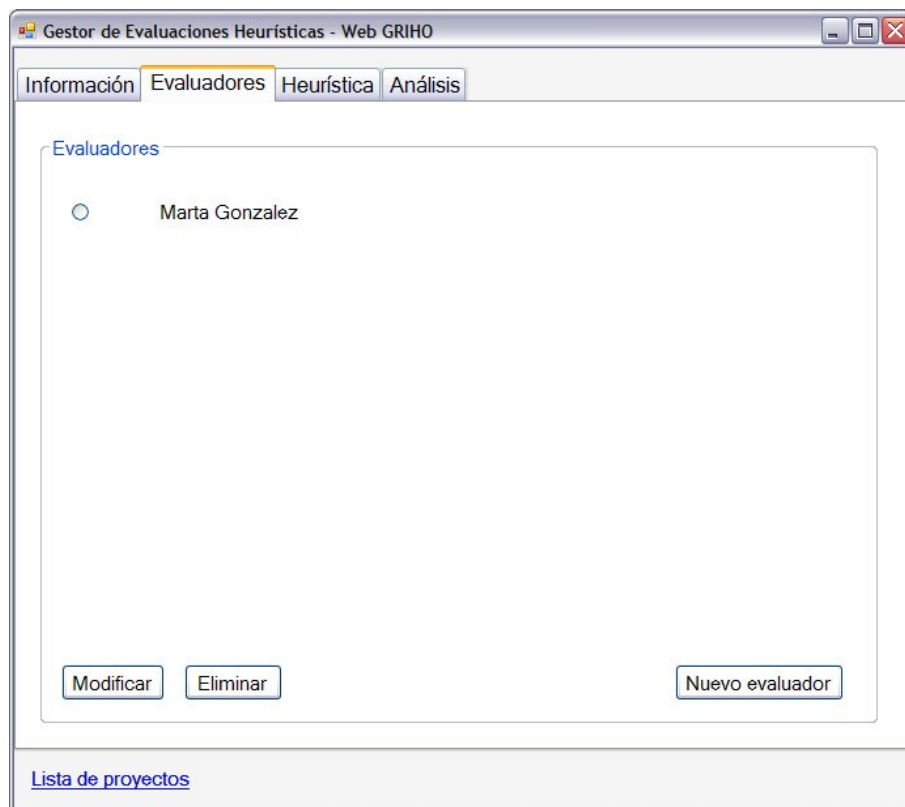
## Pestaña Evaluadores

Cuando se accede por primera vez a la pestaña *Evaluadores* aparece la pantalla vacía con un botón para registrar evaluadores.



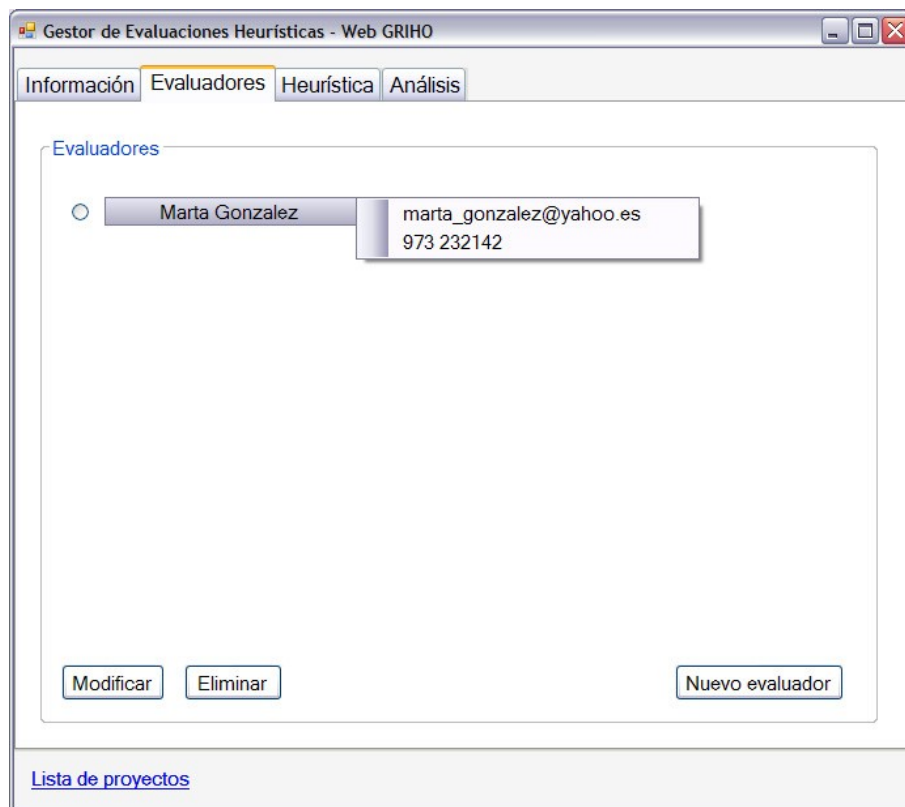
*Pantalla 5: Pestaña evaluadores vacía*

Si ya existen evaluadores registrados para este proyecto aparecerá una lista de los nombres de los mismos. También aparecerán dos nuevos botones de modificar y eliminar.



*Pantalla 6: Pestaña evaluadores*

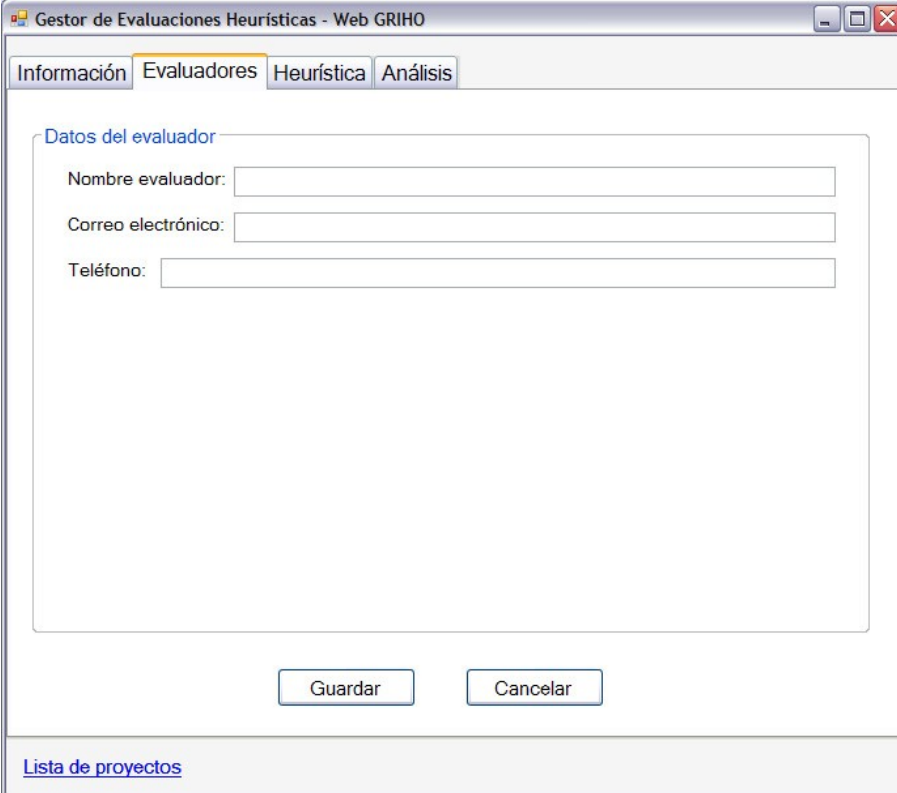
Para modificar o eliminar un evaluador, hay que seleccionar el punto relacionado con su nombre. Si se quieren ver los datos del evaluador, es necesario presionar en su nombre y aparecerá un desplegable con su información.



*Pantalla 7: Información de un evaluador*

## Registrar evaluador

Cuando se presiona el botón de *Nuevo evaluador* aparece un formulario para registrar al nuevo evaluador dentro de la misma pestaña de *Evaluadores*. Los campos del nombre y la dirección de correo electrónico del evaluador son obligatorios para realizar el registro.



Gestor de Evaluaciones Heurísticas - Web GRIHO

Información Evaluadores Heurística Análisis

Datos del evaluador

Nombre evaluador:

Correo electrónico:

Teléfono:

[Lista de proyectos](#)

Pantalla 8: Registrar evaluador

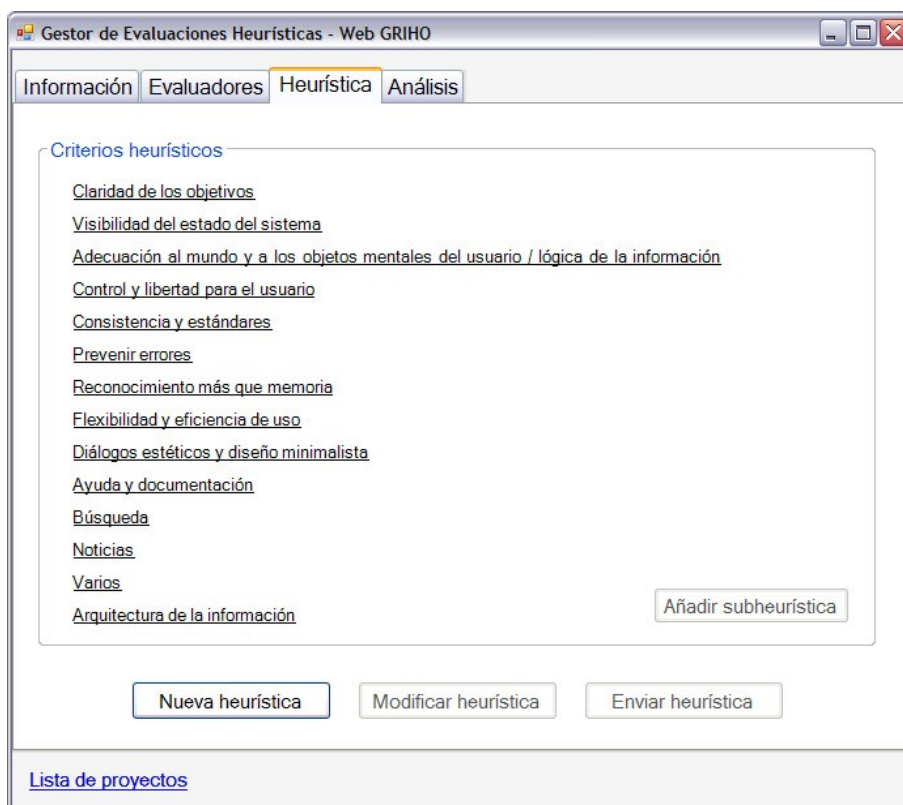
Este formulario es idéntico al formulario que aparece cuando se quieren modificar los datos del evaluador, pero mostrando por defecto la información guardada de éste.



## Pestaña Heurística

En esta pestaña se gestiona todo lo relacionado con la preparación de la heurística que se enviará a los evaluadores para que realicen la evaluación.

Al acceder a la pestaña se muestra una lista de los criterios que se utilizarán en la heurística. Estos criterios junto con sus preguntas forman parte de una plantilla definida en la aplicación.



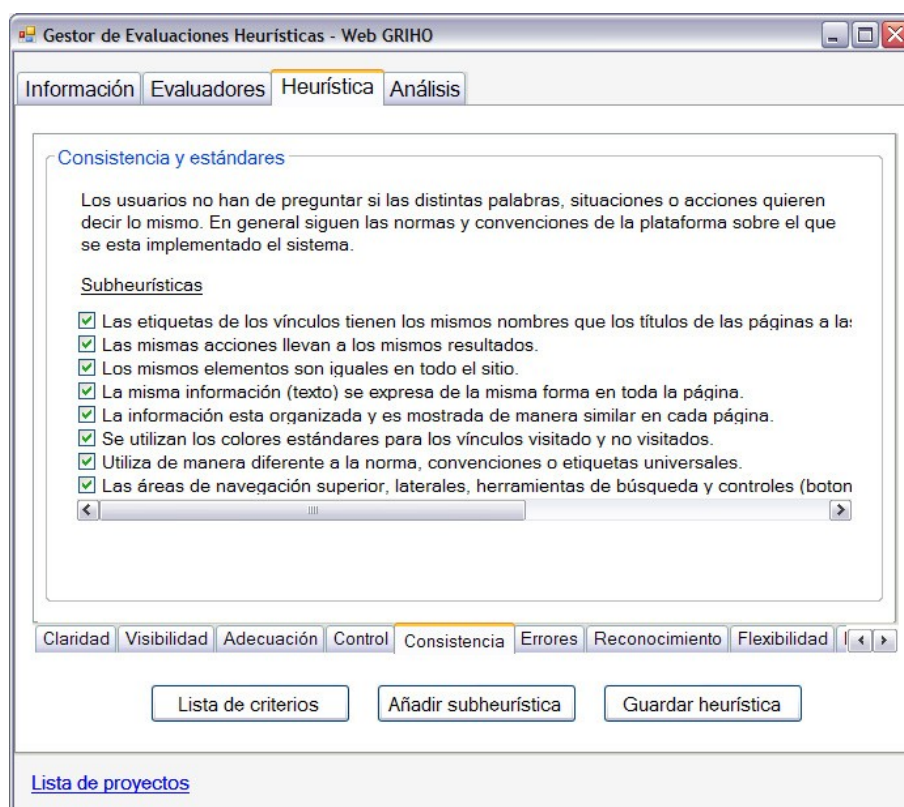
Pantalla 9: Pestaña de la heurística

Se puede crear una heurística con el botón *Nueva heurística* o presionando en uno de los criterios de la lista. El resto de botones están desactivados porque es necesario que exista una heurística para que realicen sus funciones. Cuando exista una heurística estos botones serán activados.

## Selección de subheurísticas

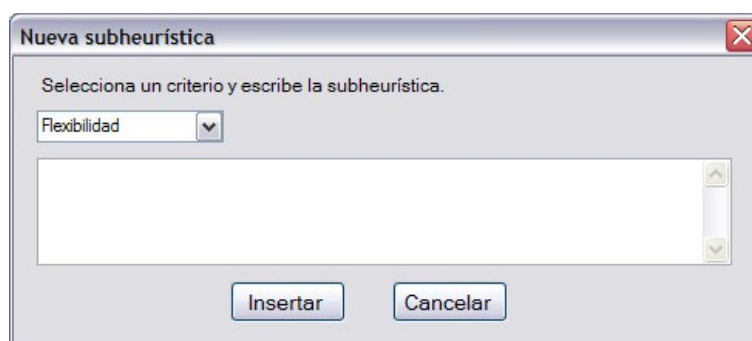
La heurística que se crea esta formada por criterios. Estos criterios aportan una serie de preguntas detalladas para la evaluación. Los criterios se muestran uno por uno mediante pestañas.

En cada pestaña de criterios se encuentra una breve descripción del criterio y una lista de subheurísticas. Todas las subheurísticas están marcadas por defecto cuando se crea una nueva heurística. El evaluador puede desmarcar las subheurísticas que no sean necesarias para la evaluación. Si se estuviese modificando una heurística ya creada, solo se mostrarían las preguntas seleccionadas por el responsable anteriormente.



Pantalla 10: Criterio de la heurística

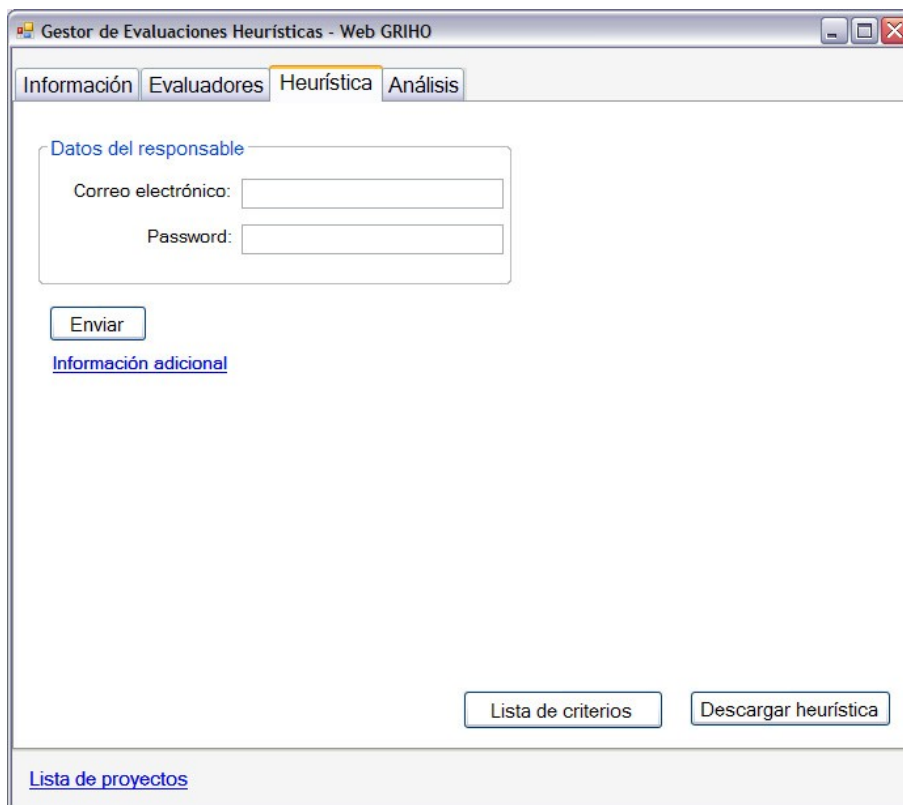
Se puede ampliar la lista de preguntas de un criterio añadiendo nuevas subheurísticas. Con el botón *Añadir subheurística* aparece una ventana adicional donde escribir la nueva pregunta.



Pantalla 11: Nueva subheurística

## Enviar heurística

Si se presiona el botón *Enviar heurística* de la pestaña *Heurística* se accede a la pantalla desde la que envía la heurística a los evaluadores registrados. Para enviar un correo es necesario introducir la dirección de correo electrónico del responsable y su contraseña.



Gestor de Evaluaciones Heurísticas - Web GRIHO

Información Evaluadores **Heurística** Análisis

Datos del responsable

Correo electrónico:

Password:

[Información adicional](#)

[Lista de proyectos](#)

Pantalla 12: Enviar correo

Cuando se envía el correo, si el gestor no contiene los datos del servidor al que pertenece la cuenta de correo, el servidor se tendrá que configurar y automáticamente aparecerá otro formulario.

Gestor de Evaluaciones Heurísticas - Web GRIHO

Información Evaluadores **Heurística** Análisis

Datos del responsable

Correo electrónico: crism1379@yahoo.es

Password: \*\*\*\*\*

Información adicional

Datos del servidor

Servidor SMTP:

Puerto:

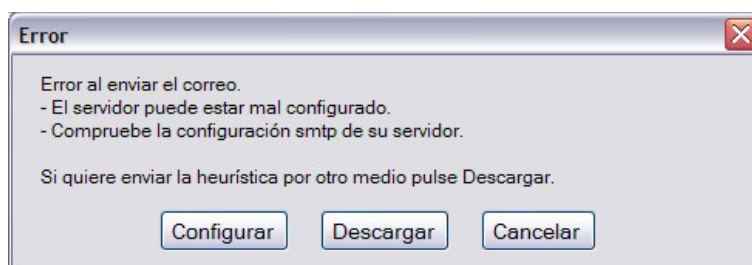
Enviar

Lista de criterios Descargar heurística

[Lista de proyectos](#)

Pantalla 13: Configuración del servidor de correo

Si al enviar un correo sucede un error, aparece un mensaje de error con las posibles soluciones que se pueden realizar.

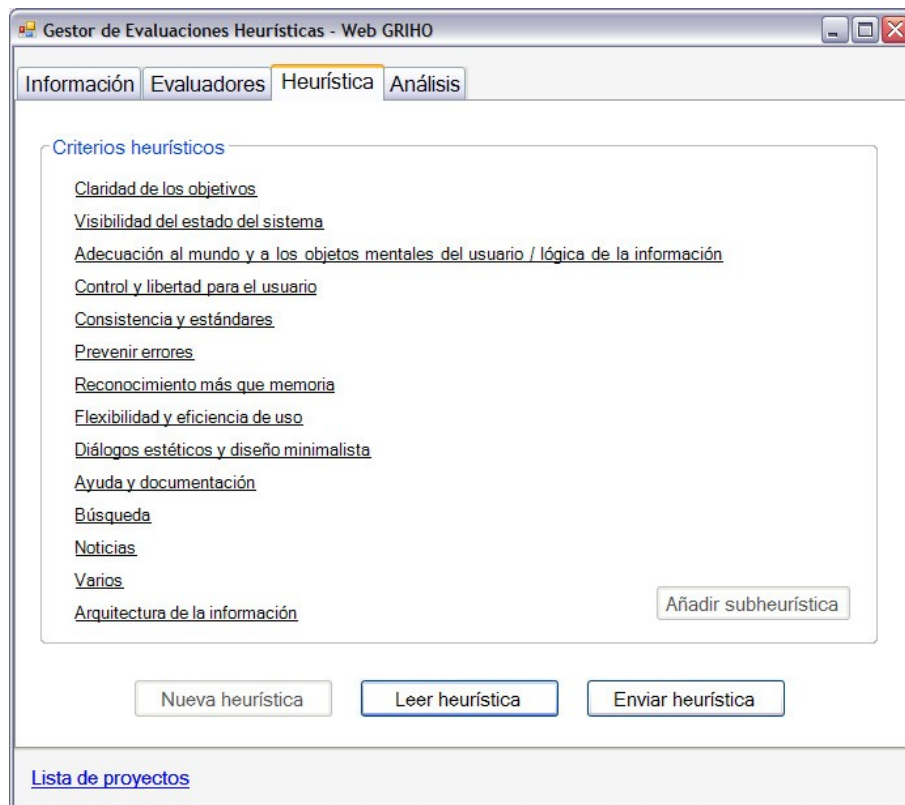


Pantalla 14: Error al enviar un correo

Si el problema no es debido a la configuración del servidor, sino del propio servidor smtp de correo, puede descargar la heurística y enviarla desde su medio de correo habitual.

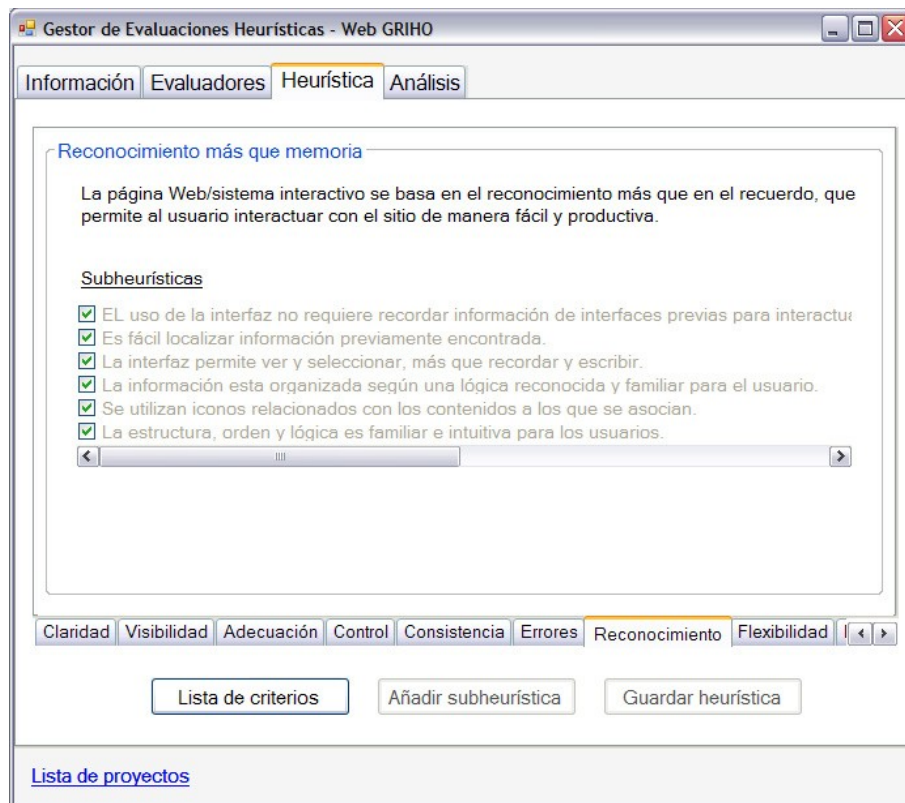
## Leer heurística

Cuando se ha enviado la heurística a los evaluadores ya no se puede modificar. Por este motivo, la pestaña de *Heurística* varía.



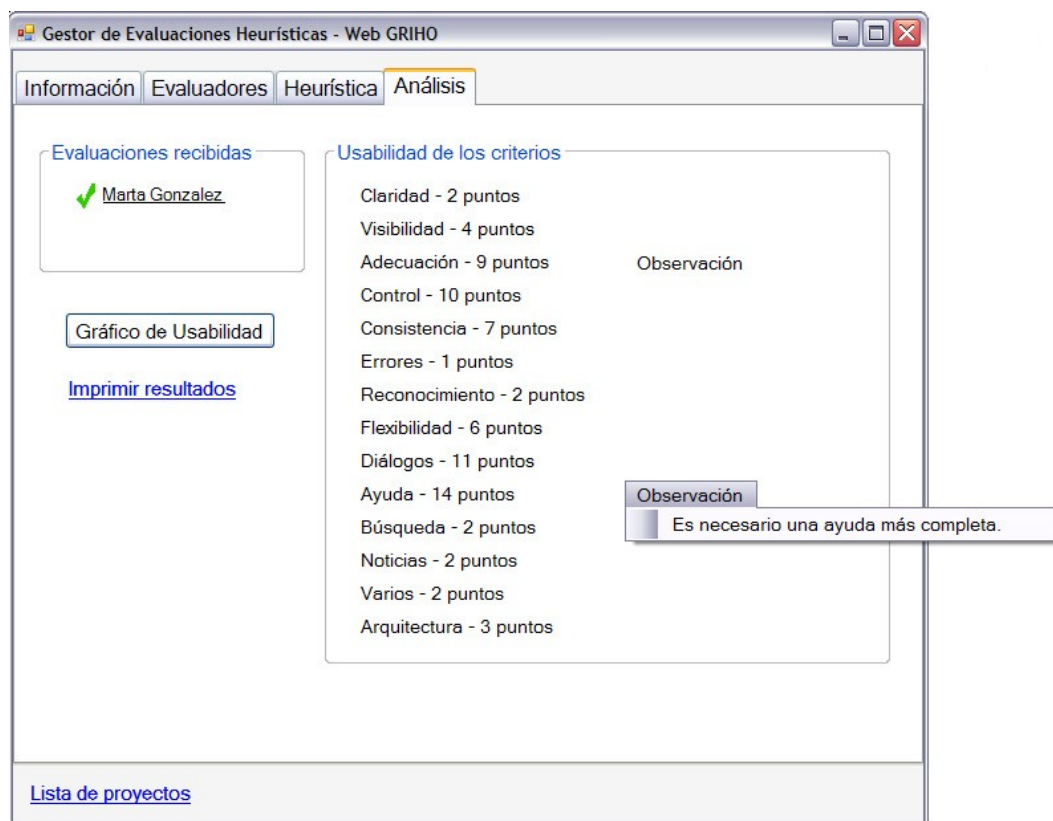
*Pantalla 15: Pestaña Heurística*

Como se puede observar, el botón que había ante para modificar, solo permite leer. La heurística se mostrará desactivada para que no puedan producir cambios.

*Pantalla 16: Leer heurística*

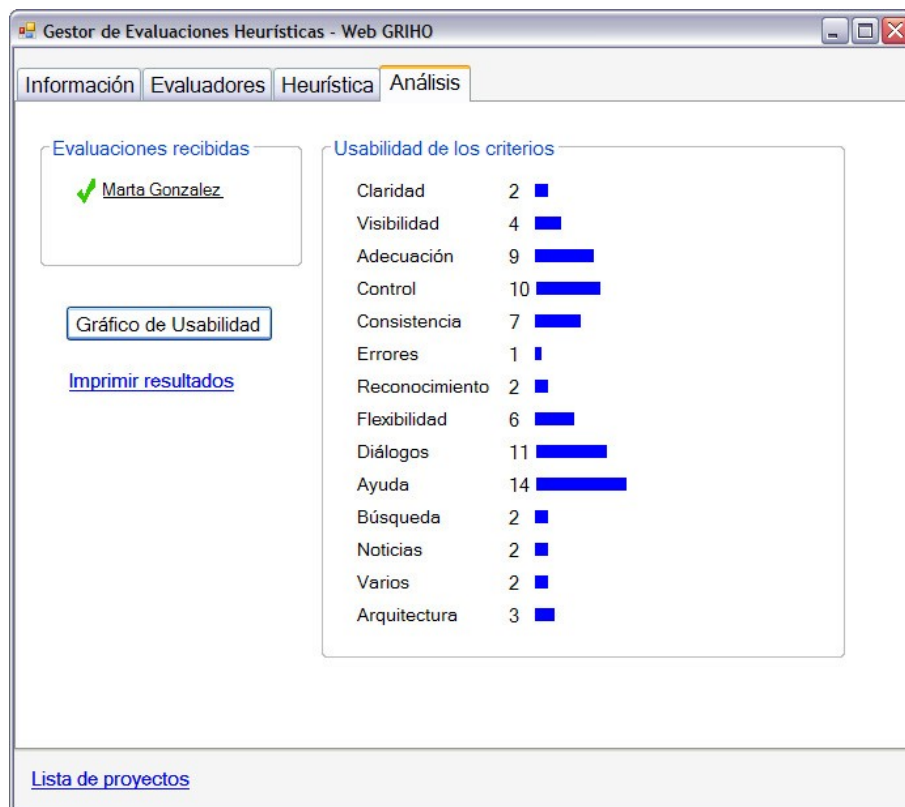
## Pestaña Análisis

El análisis de la evaluación se puede realizar cuando se ha recibido alguna evaluación. La pantalla muestra una lista de los evaluadores indicando quien ha enviado la evaluación y quien no. Si se presiona en el nombre de un evaluador que ha enviado su evaluación se muestra la puntuación media que ha recibido cada criterio con las observaciones añadidas de la evaluación de dicho evaluador.



Pantalla 17: Análisis de la evaluación de un evaluador

Presionando en el botón *Gráfico de Usabilidad* se puede ver un análisis completo de todas las evaluaciones recibidas mediante un gráfico de barras.

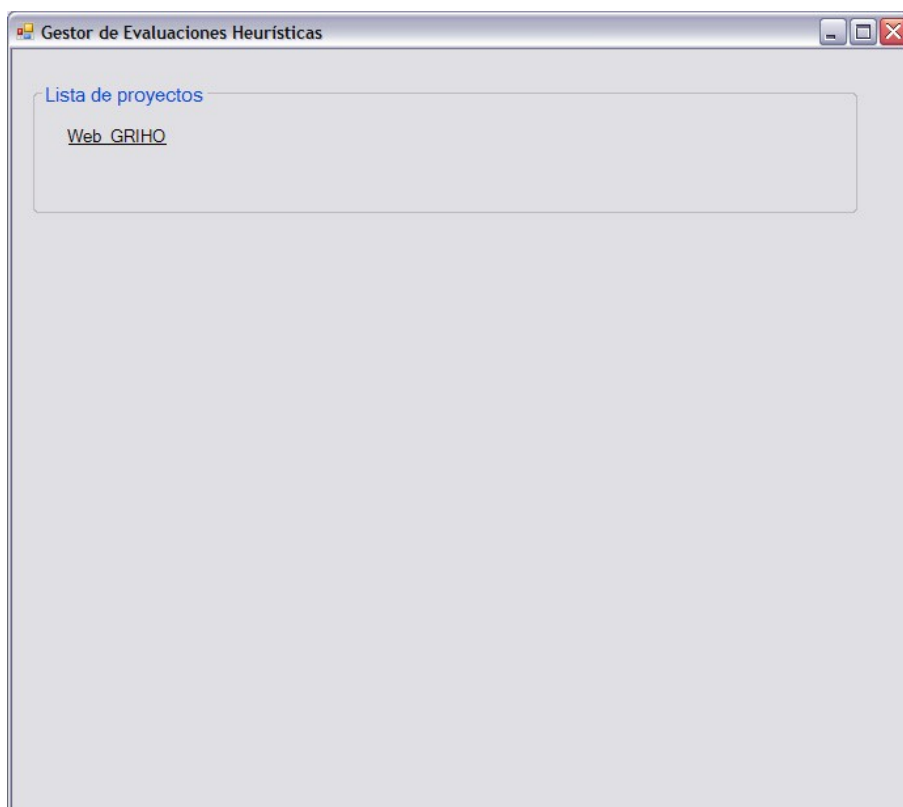
*Pantalla 18: Gráfico de usabilidad*



## Módulo del Evaluador

### Pantalla inicial

Cuando ejecutamos el módulo del Evaluador de la aplicación aparece una lista con el nombre de los proyectos que hay para evaluar. Si el gestor no contiene ningún proyecto aparece una pantalla vacía.



*Pantalla 19: Lista de proyectos*

Para acceder a cada proyecto es necesario presionar en el nombre de éste.

## Pestaña Información

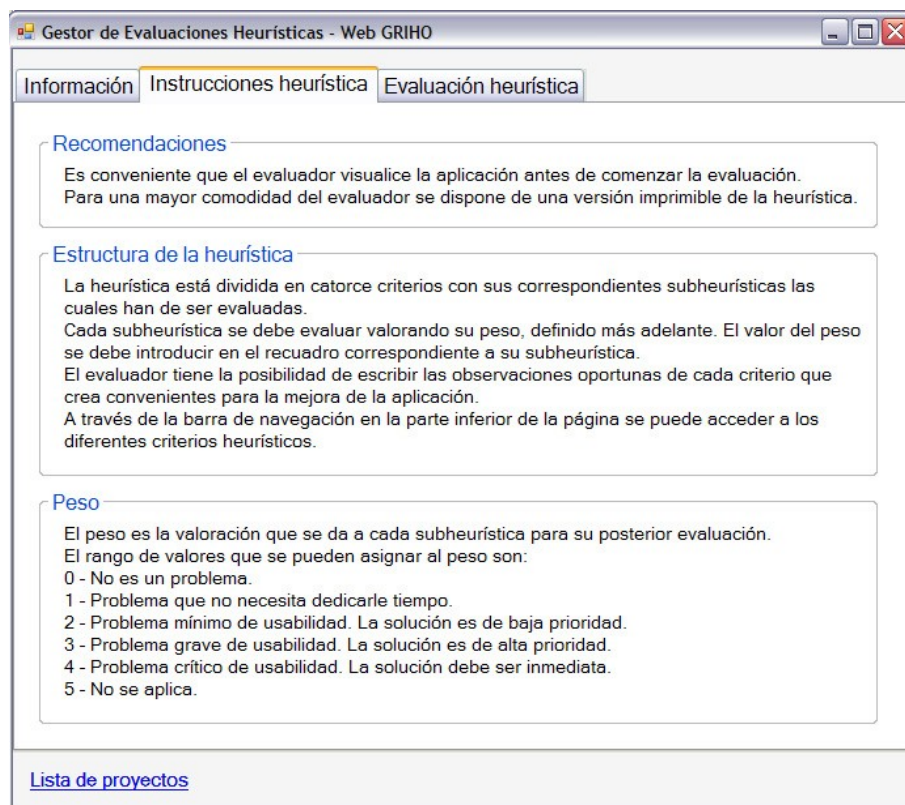
Al acceder a un proyecto, se muestran una serie de pestañas. Cada pestaña representa una actividad de la evaluación perteneciente a la tarea del evaluador. Al entrar al proyecto el gestor se sitúa en la pestaña de información del proyecto. En esta pestaña se muestra toda la información del proyecto.



*Pantalla 20: Información del proyecto*

## Pestaña Instrucciones

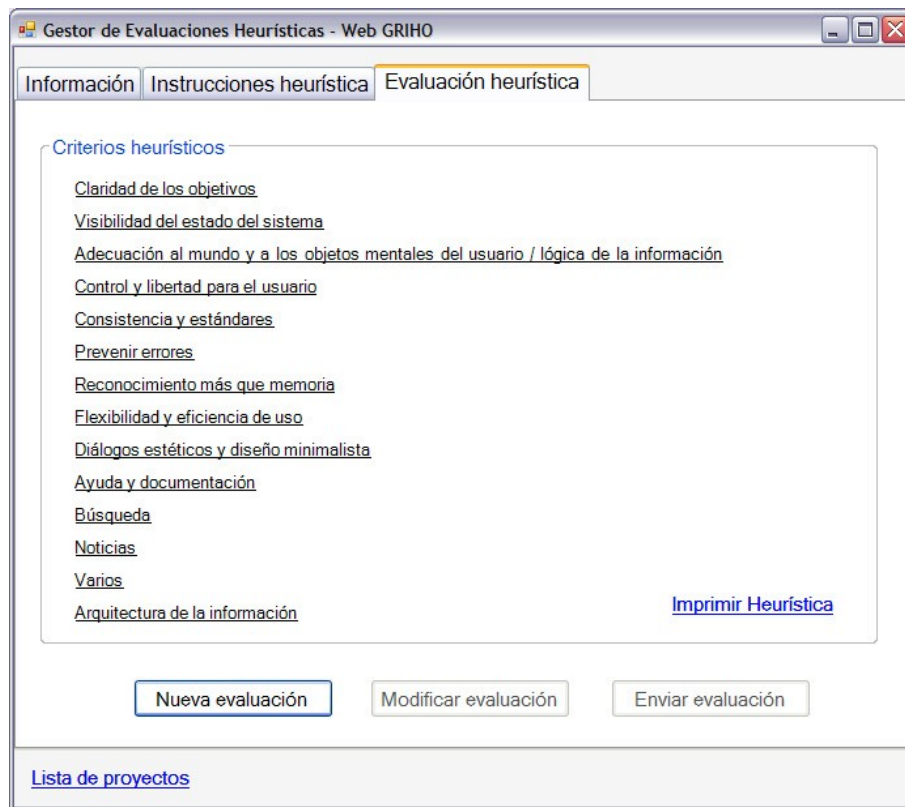
En la pestaña *Instrucciones* se detallan los pasos que debe seguir el evaluador para realizar correctamente la evaluación heurística del sistema interactivo.



*Pantalla 21: Instrucciones de la evaluación*

## Pestaña Evaluación heurística

La pantalla principal de esta pestaña es prácticamente igual que la del módulo del Responsable.



*Pantalla 22: Pestaña Evaluación*

## Realizar la evaluación

Al igual que en el módulo del Responsable los criterios se muestran uno por uno mediante pestañas.

En cada pestaña de criterios se encuentra una breve descripción del criterio y una lista de subheurísticas. Al lado de cada pregunta hay un recuadro donde introducir el grado de usabilidad que se le da a esa pregunta. Debajo de la lista hay un espacio para escribir cualquier observación de ese criterio.

Gestor de Evaluaciones Heurísticas - Web GRIHO

Información Instrucciones heurística **Evaluación heurística**

Claridad de los objetivos

La interfaz debe comunicar de manera inmediata su propósito, objetivo y funciones.

Subheurísticas

- ☐ El propósito u objetivo que la interfaz nos transmite es claro y obvio. Con una simple hojeda podemos deducir cuál es su propósito y cuál es su finalidad.
- ☐ En el caso de existir más de un objetivo, estos están relacionados con el objetivo o función global y establecen coherencia entre ellos.
- ☐ En caso de existir más de un objetivo, estos son claros y separados.

Observaciones

Claridad Visibilidad Adecuación Control Consistencia Errores Reconocimiento Flexibilidad

Lista de criterios Guardar evaluación

[Lista de proyectos](#)

Pantalla 23: Evaluar criterio

## Enviar evaluación

Este apartado es exactamente igual al del "Enviar heurística" en el módulo del Responsable.